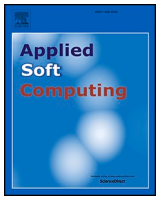
Applied Soft Computing Journal 181 (2025) 113467

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc



SMOTE-ENN resampling technique with Bayesian optimization for multi-class classification of dry bean varieties

Arnab Mukherjee [a], Mohammad Reza Chalak Qazani [b], B.M. Jewel Rana [a], Shahina Akter [a], Amirhossein Mohajerzadeh [b], Nusrat Jahan Sathi [c], Lasker Ershad Ali [d], Md. Salauddin Khan [e], Houshyar Asadi [f,*]

[a] Department of Quantitative Sciences (Mathematics), International University of Business Agriculture and Technology, Dhaka-1230, Bangladesh
[b] Faculty of Computing and Information Technology (FCIT), Sohar University, Sohar 311, Oman
[c] Department of Quantitative Sciences (Statistics), International University of Business Agriculture and Technology, Dhaka 1230, Bangladesh
[d] Mathematics Discipline, Khulna University, Khulna-9208, Bangladesh
[e] Statistics Discipline, Khulna University, Khulna-9208, Bangladesh
[f] Institute for Intelligent Systems Research and Innovation (IISRI), Deakin University, Waurn Ponds, VIC 3216, Australia

HIGHLIGHTS

- An efficient machine learning framework is designed to meet the demands of producers and customers for food security.
- Feature selection strategy is used to reduce the risk of overfitting by removing redundant features and strengthen the model predictive power.
- Hybrid resembling technique improves the model's ability to generalize to new seed or unknown seed, reducing classification bias towards the majority classes.
- Confusion matrix is followed to represents how the classification model is confused when it makes predictions. That is, it provides insight not only the errors which are made by the classifier but also types of errors that are being made.
- Feature importance is analyzed to illustrate, which types of geometric features influence the model's predictions significantly.

ARTICLE INFO

Keywords:
Feature pre-processing
SMOTE-ENN resampling technique
Light gradient boosting machine
Confusion matrix
Dry beans classification

ABSTRACT

The imbalanced classification problem poses a significant challenge in machine learning, often resulting in biased models and poor performance for minority classes. This study introduces an innovative hybrid resampling technique combining Synthetic Minority Oversampling Technique and Edited Nearest Neighbours (SMOTE-ENN), optimized using Bayesian Optimization, to address these limitations. The proposed framework integrates advanced feature pre-processing, hybrid resampling, and machine learning models to enhance classification performance. Using the publicly available dry bean dataset containing 16 geometric features of seven seed varieties, the methodology demonstrates remarkable improvements in predictive accuracy and class balance. Employing cutting-edge classifiers, the improved Light Gradient Boosting Machine (LBM) with Bayesian optimization achieved an unprecedented accuracy of 99.59 %, outperforming traditional approaches. Results reveal the potential of hybrid resampling techniques and Bayesian optimization in effectively capturing feature patterns, enhancing model diversity, and ensuring robust classification of imbalanced datasets. This research underscores the application of soft computing methods to real-world multi-class classification challenges, offering practical insights for similar domains.

1. Introduction

Data imbalance is a major challenge in data mining and machine learning, arising from a highly skewed data distribution across different classes. In such scenarios, the majority class dominates the learning process. In contrast, the minority class is often underrepresented, leading to biased models that perform poorly on rare but potentially critical cases [1,2]. This imbalance can substantially impact the performance

* Corresponding author.
E-mail address: houshyar.asadi@deakin.edu.au (H. Asadi).

https://doi.org/10.1016/j.asoc.2025.113467
Received 13 December 2024; Received in revised form 7 May 2025; Accepted 6 June 2025
Available online 18 June 2025

and generalizability of machine learning models. The impact is especially pronounced in real-life applications such as finance (e.g., fraud detection, loan default prediction), healthcare (e.g., genetic disorders, cancer detection, diabetes or heart disease onset prediction), agriculture (e.g., outbreaks of rare crop diseases or pests, classification of rare bean or fruit varieties), and rare event prediction. In these cases, the minority class holds greater practical significance. However, it lacks sufficient data for the model to learn effectively during training [3–5]. For instance, in healthcare, this issue is especially prominent in diagnostic applications, where certain diseases—such as genetic disorders or early-stage cancers—occur far less frequently than in healthy cases. This imbalance can lead standard machine learning algorithms to become biased toward the majority (healthy) class, resulting in poor sensitivity and elevated false negative rates for the minority (disease) class. Such misclassifications can have serious clinical implications, including delayed diagnoses, inappropriate treatment plans, and overlooked early interventions, compromising patient outcomes [6]. Similarly, in agriculture, detecting rare plant diseases or pest outbreaks amidst a vast majority of healthy crops requires accurate classification to avoid significant economic losses. Even without data imbalance, issues like label noise, overlapping class distributions, and other challenges may arise in typical real-world datasets. Label noise, which may result from human error or sensor inaccuracies, introduces incorrect or inconsistent annotations that distort the true decision boundaries. On the other hand, overlapping class distributions—where the feature values of different classes are not separable—complicate the learning process by increasing classification ambiguity.

In agricultural technology, imbalance classification problems like dry beans pose a dual challenge, impacting the accuracy and effectiveness of systems deployed in precision farming, automated sorting, and food quality control [7–10]. This issue can undermine precision farming's key objectives, including crop health monitoring, disease outbreak detection, and dependable yield prediction. Consistent misidentification of rare but economically or nutritionally important bean varieties may result in inadequate agronomic interventions, reducing productivity and causing financial losses. Furthermore, inaccurate classification can distort resource optimization strategies—such as targeted irrigation, fertilization, or pesticide application—resulting in inefficient input usage and potential environmental impacts. In automated sorting systems—such as those used for grading beans based on size, shape, and quality—imbalanced data can adversely affect the overall efficiency and profitability of the sorting process. This may lead to errors in sorting out defective or lower-quality beans. Moreover, the importance of accurate classification cannot be overstated in food quality control, as it directly influences consumer safety and satisfaction. The challenge here is that imbalanced data may lead to the failure to detect less common but critical quality issues, such as slight defects or contamination, which are essential for meeting regulatory standards. This can result in compromised product quality, potentially leading to financial losses, consumer dissatisfaction, or health risks.

Despite continuous advancements in research over the past decades, learning from data with imbalanced class distributions remains a compelling and challenging study area. Earlier approaches to dry bean classification exhibit several limitations at both the data mining and machine learning stages [11,12]. For instance, undersampling methods such as NearMiss and Edited Nearest Neighbour (ENN) discard potentially informative samples from the majority class. This leads to a loss of data diversity and increased sensitivity to small changes. Conversely, oversampling methods like Random Over-Sampling (ROS) duplicate minority class instances, which can result in overfitting. This increases training time and computational costs and risks the model memorizing minority instances rather than learning generalizable patterns. Consequently, the lack of additional variance reduces its ability to generalize to unseen data. Other widely used oversampling techniques include the Synthetic Minority Over-sampling Technique (SMOTE) and its variants, such as Borderline-SMOTE, ADASYN, and Safe-Level-SMOTE. However, these methods often generate duplicate samples, ignore underlying data distributions, introduce potentially inaccurate instances, perform poorly with high-dimensional data, and are highly sensitive to noise.

Additionally, they may distort the natural distribution of geometric features and increase model complexity without improving generalization. At the data mining stage, few studies have investigated correlations among geometric features, and many overlook the elimination of highly correlated attributes (e.g., perimeter, convex area, major axis length, and minor axis length). This oversight can lead to classification confusion and reduced accuracy by redundantly capturing similar bean shape or size aspects. Finally, existing approaches lack automated hyper-parameter tuning and fail to provide insights into feature importance or identify which geometric features significantly influence the model's predictions.

Specific challenges still need to be addressed for real-life applications of imbalanced classification problems, such as data mining, appropriate feature extraction or selection, and reliable classifier performance. Considering these challenges, this work contributes to imbalanced multi-class classification in agricultural datasets. The key topics covered are summarized as follows:

1. Imbalanced learning aims to design intelligent systems capable of robustly tackling data distribution bias, enabling learning algorithms to handle imbalanced data more efficiently.
2. It introduces a hybrid SMOTE-ENN resampling strategy to mitigate class imbalance, particularly enhancing minority class representation effectively.
3. It integrates Bayesian Optimization for automated and efficient hyper-parameter tuning, reducing the need for manual intervention while improving model performance.
4. The proposed framework is rigorously evaluated on the real-world multi-class dry bean dataset, demonstrating its practical applicability and robustness.
5. A combined approach of feature selection and Principal Component Analysis (PCA) is employed to address feature redundancy and improve computational efficiency.
6. Finally, SHAP (SHapley Additive exPlanations) analysis ensures model interpretability, offering insights into feature contributions and supporting transparent decision-making.

In this work, Fig. 1 illustrates a complete machine-learning pipeline for classifying dry bean varieties. The process begins with inputting raw dry bean data and then extracting geometric features, which undergo pre-processing to ensure data quality and consistency. A resampling technique is applied before splitting the data into training and testing sets to address class imbalance. A learning model is trained and optimized through Bayesian optimization to fine-tune its hyper-parameters. The model generates predictions for dry bean classes, and its performance is evaluated using a colour-coded confusion matrix that highlights classification accuracy across the different bean types.

The article is organized as follows: **Section 2** presents an extensive literature review. **Section 3** describes the necessary materials and methods. Cutting-edge ML techniques with parameter optimization are discussed in **Section 4**. **Section 5** presents experimental results, graphical discussions, and comparative studies. The article concludes with **Section 6**.

2. Literature reviews and motivations

This section overviews exploratory data analysis-based multiclass classification learning models using various data imbalance techniques. Most of the conventional machine learning models are designed to perform on balanced data with roughly equal sample sizes between different classes. From the perspective of applications, various types of features such as colour, shape, texture, diagnostic, physical, meteorological and morphological features are utilized when training machine

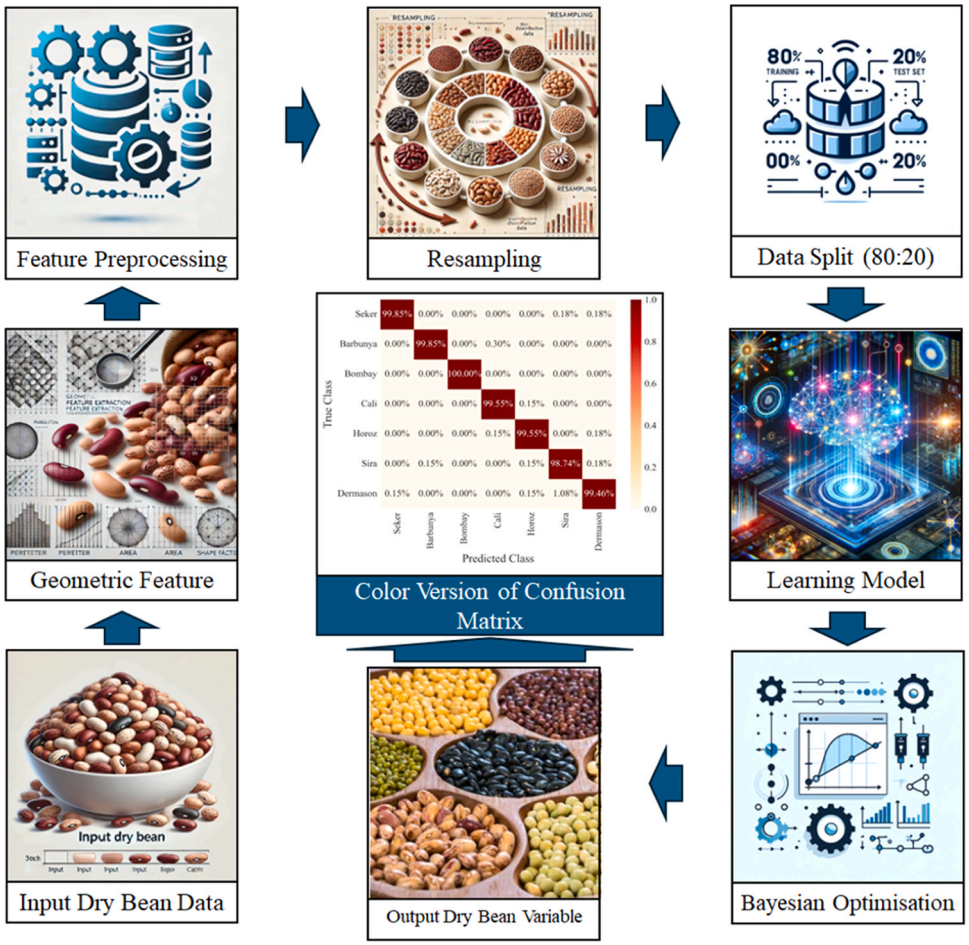


Fig. 1. The graphical abstract of the proposed methodology.

learning models.

Firstly, Koklu et al. [7] introduced an automated computer vision system for dry bean classification into genetic varieties using geometric and shape features. Debjit et al. [13] designed a healthcare monitoring system to detect COVID-19 using Harris Hawks's optimization and diagnostic features. Islam et al. [14] proposed a predictive model for forecasting the chances of cesarean or C-section (CS) delivery using Henry gas solubility optimization. Numerous factors contribute to the risk of heart disease, highlighting the urgent need for accurate and efficient diagnostic methods. In response, Subathra and Sumathy [15] introduced a novel framework using Bolstered-up Beetle Swarm optimization based feature selection to enhance heart disease detection performance. Alam et al. [16] explored credit card default prediction by using ***k***-means SMOTE oversampling to enable proactive risk mitigation, emphasizing the importance of accurate forecasting in financial decision-making. Almost all of their frameworks [7,13–16] used traditional ML models such as Logistic Regression (LR), Extreme Gradient Boosting (XGB), Random Forest (RF), Dense Stacking Ensemble (DSE), Weighted Ensemble Learning (WEL) and Gradient Boosted Decision Tree (CAT), etc. based on raw features. The model performances fluctuate between 88.3 % and 96.98 % to their corresponding experimental imbalanced datasets. However, the trade-off between accuracy and efficiency has become the key challenge in realistic situations like medical sciences due to having more chances of fault predicting.

In addition, lack of interpretability on unknown cases, limited data access to learn, expensive computations, etc., are the drawbacks of traditional ML algorithms. Deep learning is an advanced subset of machine learning for these tasks to overcome these obstacles and drawbacks. In, [8] Taspinar et al. proposed pre-trained CNN-based transfer learning approaches, namely InceptionV3, VGG16, and VGG19 structures, for feature extraction and classification operations. In later research, the same authors [9] developed an ELM-based salp swarm optimisation algorithm (SSA) using image features via GoogLeNet transfer learning. The success rates of InceptionV3 DL and SSA-based ELM models were 84.48 % and 83.71 % on the haricot bean dataset consisting of 33,064 images of 14 genetic varieties. Fahim et al. [17] developed a highly accurate classification model for distinguishing genetic variations among dry bean varieties. The custom CNN model achieved an impressive accuracy of 99.85 % with high computational cost. The Xception and MobileNet models showed slightly lower accuracy at 82 %. These DL models cannot classify the complex patterns in the imbalanced data due to the lack of hypermeter optimization, and they are becoming more complicated. However, the decision boundaries defined by the learning algorithms exhibit bias toward the majority class due to having imbalanced data within different classes. The problem of imbalanced data classification presents numerous challenges that have been extensively studied in [18–20] like this style [12–16]. Several sampling algorithms can be applied in different contexts to address these issues, depending on the dataset's characteristics and learning objectives. Khan et al. [10] focused on outlier removals and distinguished classification performance between balanced and imbalanced datasets using dry beans. With an accuracy rate of 95.40 %, the XGB model outperformed the other methods with the help of the Adaptive Synthetic (ADASYN) oversampling algorithm.

Macuácua et al. [21] followed data mining and augmentation techniques to improve the classification outcomes. The overall accuracies were 92.4 % without any feature pre-processing, 92.8 % with hyper-parameter optimization, and 95.9 % using original data,

optimized hyper-parameters, and Synthetic Minority Oversampling Technique (SMOTE) balancing, respectively. There was a significant increase of around 2.6 % in the KNN model. Feizi et al. [22] produced a data dictionary that prioritizes samples based on their importance within each manifold, incorporating weighted manifold scores and ***k***-nearest neighbours. It identifies the significance of synthetic data generation using a linear combination of multiple manifolds, leveraging the data's inherent substructures. Djafri [23] introduced PRO-SMOTE, an extension of SMOTE that reduces majority classes and optimally increases the minority class based on conditional probabilities. PRO-SMOTEBoost improved performance by 10–40 % but may risk reducing dataset quality by incorrectly removing or failing to identify the right samples.

A framework for multi-class imbalanced big data on Spark is proposed by Sleeman and Krawczyk [24], focusing on balancing classes by analysing instance-level difficulties. Though effective, the model's reliance on Spark infrastructure and specific resampling strategies may limit its flexibility and adaptability in non-Spark environments. Shirvan et al. [25] designed deep generative models to address multiclass imbalanced data problems by using Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs). Each of the models discussed has unique strengths and limitations in terms of scalability, training complexity, and susceptibility to bias. These factors should be evaluated based on the nature of the data and the specific application. For instance, GANs often struggle to capture the full diversity of the data due to issues like mode collapse, which leads to reduced variation in generated outputs. Borowska and Stepaniuk [26] developed a rough-granular approach (RGA) for selective oversampling and neighbourhood editing. This method effectively handled imbalanced data by focusing on specific regions of the feature space, reducing misclassification rates. Ng et al. [27] introduced a cost-sensitive localized generalization error model (c-LGEM) to prioritize minority class accuracy. This approach effectively balanced error across classes and demonstrated superior performance on multiple datasets. Zheng et al. [28] introduced a genetic algorithm to optimize sampling ratios, enhancing classification stability. However, the approach's reliance on algorithmic fine-tuning may limit its effectiveness on datasets with highly dynamic or evolving imbalances. Doan et al. [29] proposed a cluster-based data splitting technique (WICS) combined with SMOTE-NC for impact damage classification in imbalanced datasets. Their results highlighted improvements in classification performance, achieving stability and precision in small, imbalanced datasets. An equalization ensemble method [30] uses an equalization under-sampling scheme and weighted integration to improve classification accuracy in large, imbalanced datasets. Despite its strengths, the model's dependency on under-sampling can lead to a loss of valuable majority-class information, potentially affecting accuracy in cases of extreme imbalance. Dixit and Mani [31] addressed noise and borderline examples in imbalanced data using the SMOTE-TLNN-DEPSO method. Their hybrid approach optimized noisy samples rather than removing them, maintaining class balance and enhancing classification accuracy.

Recently, Pepsi et al. [32] addressed the challenges of non-stationary data with class imbalance using a Hybrid Firefly Optimization algorithm and an oversampling technique to enhance minority class representation. However, the model's performance may be limited in real-world scenarios with highly dynamic data streams requiring rapid adjustments. Alex et al. [33] proposed the GA-SMOTE-DCNN technique, which integrates a genetic algorithm for feature selection, SMOTE for oversampling, and a deep convolutional neural network for classification. Their study demonstrated significant accuracy improvements, highlighting the scalability of the approach across high-dimensional and imbalanced data classification problems. Using federated learning, Liu et al. [34] introduced a privacy-preserved hotel customer classification model. By incorporating an attention mechanism and a dynamic client selection strategy, their model effectively balanced global and local performance, enhancing accuracy and preserving privacy in imbalanced datasets. Kamro et al. [35] presented a metaheuristic-driven space partitioning and ensemble learning framework, which combines SMOTE with space partitioning to create balanced subspaces. Their method outperformed state-of-the-art approaches, offering a promising solution for minimizing alterations to the original data distribution. Other methods do not rely solely on synthetic sampling techniques. Instead, they explore alternative strategies for selecting data mining stages or classification algorithms to suit the characteristics and requirements of imbalanced datasets.

Despite these challenges, addressing class imbalance is crucial for advancing agricultural technology and promoting more sustainable & efficient agricultural practices. The aforementioned techniques have shown significant potential in tackling imbalanced classification problems. The novelty of this work lies in incorporating a hybrid resampling method, feature selection, cost-sensitive learning models, and automated hyper-parameter tuning to enhance classification performance on minority classes without compromising overall accuracy. More specifically, the hybrid resembling technique integrates SMOTE and Edited Nearest Neighbour (ENN) to reduce the trend of bias towards the majority class and increase model diversity. Feature correlation analysis and selection are performed to strategically reduce inter-class ambiguities and enhance the model's predictive power, interpretability, generalization, and computational efficiency. Furthermore, cost-sensitive learning models with automated hyper-parameter tuning can be effective alternatives to traditional machine learning and deep learning techniques for high-dimensional, large-scale classification problems, data mining, and related tasks.

3. Materials and methods

This section provides an overview of the proposed machine learning approach, which incorporates two schemes: geometric feature concatenation and robust classifier selection. Outliers may affect each bean's dimensional and shape features on different scales. These unnecessary outliers will be removed to prevent deviation in predictions. At the same time, the remaining features will be standardized into uniform numeric values within a fixed range, ensuring consistency and conformity. In this work, SMOTE generates a set of synthetic instances without considering the proximity of the majority class, which may lead to overlapping or ambiguous class boundaries. Then, the ENN technique is applied to remove instances that differ from most of their ***k***-nearest neighbours, aiming to improve generalization. The experimental dataset was split arbitrarily into training and testing sets to train the ML models. The ML classifiers' influential hyper-parameters are selected, and their optimal values are explored using the Bayesian optimization technique and statistical evaluations. **Fig. 2** illustrates the proposed framework using exploratory data analysis.

3.1. Dataset

For multiclass seed classification, Koklu et al. [7] collected a dataset of dried beans under varying imaging conditions from various cultivation sites in Turkey. The collection includes 13,611 images of 7 varieties of dry beans. The geometric features consist of 12-dimensional and 4 shape-related features for each bean, which were measured after seed segmentation using Otsu's global thresholding method. This study employs geometric features due to their strong ability to capture the structural and morphological characteristics of the investigated objects. These features are often highly discriminative, especially when dealing with small variations in appearance between different types of object varieties. As a result, this makes them particularly useful for classification tasks where other features might not capture the subtle differences. Compared to purely texture- or colour-based descriptors, geometric features exhibit a high degree of robustness against lighting, background, and imaging conditions variations. They are often invariant to transformations like rotation, scaling, and translation, enhancing the

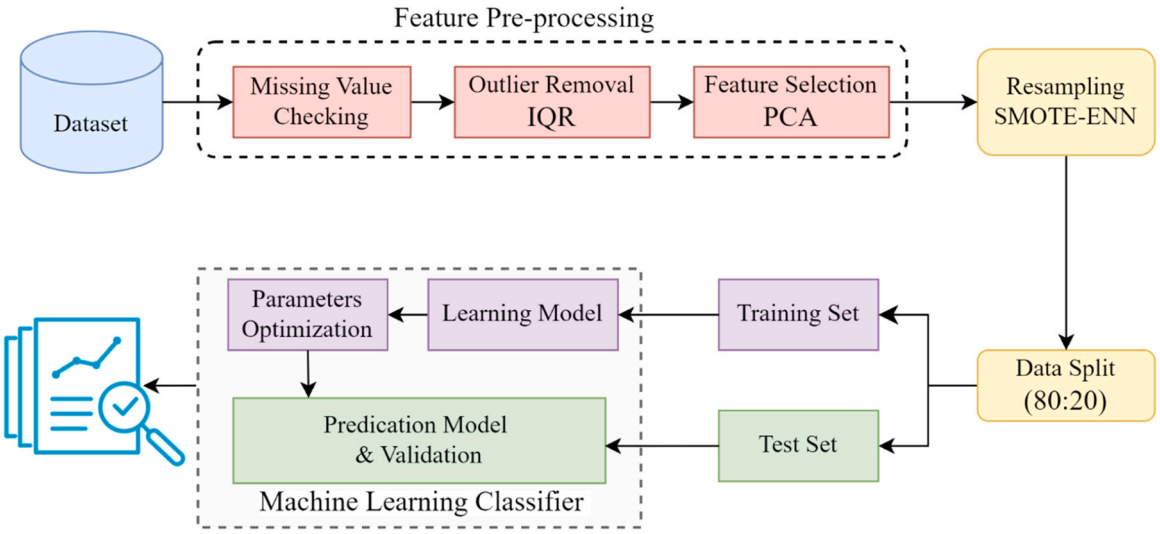


Fig. 2. The architectural framework of the proposed dry bean classification model.

model's generalizability across different imaging setups. Since the geometry of an object is closely tied to its biological or physical properties, incorporating these features allows for a deeper understanding of underlying patterns and differences between object varieties. Therefore, including geometric features was essential for improving both the reliability and the interpretability of the model outcomes in this research.

Weight variances are owing to the weight variations in seed varieties since the instances were obtained from the identical weight of each variety. The number of observations (pieces) and weight in terms of average gram per seed for each variety are listed in Table 1. The genetic varieties of dry beans are defined as the target classes: Bombay, Seker, Barbunya, Dermason, Cali, Sira, and Horoz. Their raw dataset was collected from the UCI ML Repository Website, https://archive.ics.uci.edu/ML/datasets/Dry+Bean+Dataset.

The grains yielded a dataset of 16 characteristics, including 4 shapes and 12-dimension forms. The spatial features are 1) Area ($\boldsymbol{A}$): Area counts the number of pixels within the bean region and its boundaries; 2) Convex area ($\boldsymbol{C}$): The area of a bean that belongs to the smallest convex polygon; 3) Perimeter ($\boldsymbol{P_m}$): The length of a bean's border is its perimeter; 4) Major axis length ($\boldsymbol{L}$): The distance of a bean's longest line measured horizontally between its ends; 5) Equivalent diameter ($\boldsymbol{E_d}$): It defines the diameter of a circle with a similar area of bean seed, i.e., $\boldsymbol{E_d} = \sqrt{(4\boldsymbol{A}/\pi)}$; 6) Minor axis length ($\boldsymbol{l}$): The distance of a bean's longest line that is measured vertically concerning the main axis; 7) Aspect ratio ($\boldsymbol{A_r}$): The ratio of major axis length ($\boldsymbol{L}$) to minor axis length ($\boldsymbol{l}$), i.e. $\boldsymbol{A_r} = \boldsymbol{L}/\boldsymbol{l}$; 8) Eccentricity ($\boldsymbol{E_c}$): Eccentricity refers to how round or flat a bean's shape is; 9) Solidity ($\boldsymbol{S}$): The ratio of the area ($\boldsymbol{A}$) to the convex area ($\boldsymbol{C}$), i.e., $\boldsymbol{S} = \boldsymbol{A}/\boldsymbol{C}$. It is also defined as convexity. 10) Roundness ($\boldsymbol{R}$): $\boldsymbol{R} = 4\pi\boldsymbol{A}/\boldsymbol{P_m^2}$; 11) Extent ($\boldsymbol{E_x}$): The ratio of the area ($\boldsymbol{A}$) to the bounding box ($\boldsymbol{B_b}$) i.e. $\boldsymbol{E_x} = \boldsymbol{A}/\boldsymbol{B_b}$; 12) Compactness ($\boldsymbol{C_o}$): The ratio of equivalent diameter ($\boldsymbol{E_d}$) to major axis length ($\boldsymbol{L}$), i.e., $\boldsymbol{C_o} = \boldsymbol{E_d}/\boldsymbol{L}$; 13) ShapeFactor1: The ratio of major axis length ($\boldsymbol{L}$) to the area ($\boldsymbol{A}$), i.e., $\boldsymbol{SF_1} = \boldsymbol{L}/\boldsymbol{A}$; 14) ShapeFactor2: The ratio of minor axis length ($\boldsymbol{l}$) to the area ($\boldsymbol{A}$), i.e., $\boldsymbol{SF_2} = \boldsymbol{l}/\boldsymbol{A}$; 15) ShapeFactor3: $\boldsymbol{SF_3} = 4\boldsymbol{A}/\pi\boldsymbol{L_2}$. 16) ShapeFactor4: $\boldsymbol{SF_4} = 4\boldsymbol{A}/\pi\boldsymbol{lL}$. The statistical measures, such as minimum (Min.), maximum (Max.), mean, and standard deviation (Std. deviation), of the geometric features are obtained from all the dry bean instances. There exist many inconsistencies among the feature distribution that can reduce normality. For instance, Eccentricity to ShapeFactor4 has minimum values of less than 1, and the values of ShapeFactor1 & ShapeFactor2 are negligible compared to others. On the other hand, the feature values of Area & Convex Area are too high. That is, they can cause bias and/or influence estimates. However, the geometric measures are randomly distributed due to being tiny, which makes the feature distribution more complicated. The statistical measures concerning geometric features are listed in Table 2.

Table 1
The distribution of dry bean varieties.

No.	Name of Varieties	Piece	Weight
1	Bombay	522	1.92
2	Seker	2027	0.49
3	Barbunya	1322	0.76
4	Dermason	3546	0.28
5	Cali	1630	0.61
6	Sira	2636	0.38
7	Horoz	1928	0.52
	Total	13,611	4.96

3.2. Data pre-processing

The data pre-processing approach detects and eliminates the null values and outliers using a boxplot and the interquartile range (IQR) of the Python program. The first quartile $\boldsymbol{Q_1}$ & third quartile $\boldsymbol{Q_3}$ are determined by the midpoint method, and then the IQR is computed by the following:

$$IQR = Q_3 - Q_1 \tag{1}$$

In the presence of outliers, standardisation can become skewed or biased, which leads to several issues like slow convergence, unstable gradients, difficulty setting hyper-parameters, deviation from the model's prediction and numerical issues. The dry bean dataset contains outliers in most geometric features except ShapeFactor2, as shown in Fig. 3.

The vertical line displays the first quartile($\boldsymbol{Q_1}$) and third quartile($\boldsymbol{Q_3}$), arranged from bottom to top. In contrast, the blue horizontal line within IQR denotes the median. The bounds of the upper quartile ($\boldsymbol{Q_3}$) and lower quartile($\boldsymbol{Q_1}$), are calculated with ($\boldsymbol{Q_3}$+1.5IQR) and ($\boldsymbol{Q_1}$−1.5IQR). The vertical dots outside the upper and lower bounds are known as outliers, illustrated in Fig. 4.

Removing outliers helps standardisation rescale the given features with unit variance and zero means. The advantage of StandardScaler over normalisation is that it just scales and alters the distribution of each feature, not changing its form [36]. Consequently, each geometric measure $\boldsymbol{x_{i,n}}$ of the input features is converted into standard score $\boldsymbol{z_{i,n}}$ to make the ML process more efficient and effective with the help of:

Table 2
Geometric feature description of dry bean varieties in pixels.

Serial No.	Geometric Features	Min.	Max.	Mean	Std. deviation
1	Area	20420.00	254616.00	53048.28	29324.09
2	Convex Area	20684.00	263261.00	53768.20	29774.91
3	Perimeter	524.74	1985.37	855.28	214.28
4	Major Axis Length	183.60	738.86	320.14	85.69
5	Equiv Diameter	161.24	569.37	253.06	59.17
6	Minor Axis Length	122.51	460.19	202.27	44.97
7	Aspect Ratio	1.025	2.430	1.583	0.247
8	Eccentricity	0.219	0.912	0.751	0.0927
9	Solidity	0.919	0.995	0.987	0.005
10	Roundness	0.489	0.991	0.873	0.059
11	Extent	0.555	0.866	0.749	0.049
12	Compactness	0.641	0.987	0.799	0.062
13	ShapeFactor1	0.003	0.011	0.007	0.001
14	ShapeFactor2	0.001	0.004	0.002	0.001
15	ShapeFactor3	0.410	0.975	0.644	0.099
16	ShapeFactor4	0.948	0.999	0.995	0.004

$$z_{i,n} = \frac{x_{i,n} - \mu_i}{\sigma_i} \tag{2}$$

Where, $\boldsymbol{\sigma_i}$ and $\boldsymbol{\mu_i}$ are the standard deviation and mean of data, respectively.

3.3. Feature selection strategy

The following key criteria and factors are considered in the feature selection stage to ensure a robust and generalizable model. The objective is to ensure that features are strongly correlated with the target variable but minimally correlated with each other. This can occur especially when dealing with subtle differences in appearance between various types of objects, such as dry beans. Firstly, correlation measurement among features is measured to identify which features are most informative and relevant to the target variable. Highly correlated features can introduce redundancy, leading to multicollinearity and model complexity. Secondly, the predictive power of each feature is assessed, with weak or irrelevant features being excluded to prevent them from contributing noise to the model. Thirdly, a dimensionality reduction technique projects the data into lower-dimensional spaces, preserving variance while reducing overfitting.

Additionally, the consistency of features across training and testing sets is examined, ensuring that selected features generalize well to unseen data. This work adopts correlation-based feature selection to prevent the model from overfitting or underfitting. The correlation among features of dry beans is visualized through a heatmap and correlation matrix in Fig. 5a, which shows that the "Perimeter, ConvexArea & EquivDiameter"; "Major Axis Length & Minor Axis Length" and "Eccentricity & AspectRation" features with high correlation are more linearly dependent. These convey almost similar information regarding the target variable. Therefore, only one representative feature from each correlated group should be retained to avoid multicollinearity and reduce model complexity. The concept of correlation between more features is not addressed in the literature. It is more convenient to drop features that make no significant contribution to the model's predictions but increase model complexity without adding value. Here, the selected correlated geometric features, such as "Perimeter, Area, ConvexArea, EquivDiameter", "ShapeFactor3, Eccentricity, AspectRatio, Compactness", and "MinorAxisLength, ShapeFactor1" are projected using dimensionality reduction techniques like PCA, as described in [21]. The resulting components, namely PCA1, PCA2, and PCA3, are used instead of the original features, as seen in Fig. 5b. Finally, these components are concatenated with the rest of the geometric features to construct a discriminative feature set. With careful consideration of these criteria and domain knowledge, the correlation-based feature selection process helps build a robust prediction model that is accurate and well-generalizable to unseen data. With careful consideration of these criteria and domain knowledge, the correlation-based feature selection process helps build a robust prediction model that is accurate and well-generalizable to unseen data.

3.4. Hybrid resampling technique

There are diverse resampling techniques to handle imbalanced datasets and improve the model's interpretability, generalizability and predictive accuracy. Nonetheless, no resampling technique has an apparent advantage over another. This study proposes a SMOTE-ENN-based hybrid resembling technique to minimise bias towards the majority class of dry beans [37]. The working procedure of the SMOTE-ENN technique is explained mathematically and graphically below.

SMOTE algorithm generates synthetic instances by evaluating neighbour instances using linear interpolation for the minority class. For every minority instance $\boldsymbol{x_i}$, SMOTE firstly selects its $\boldsymbol{k}$-nearest neighbours $\boldsymbol{k_{x_i}}$ from the minority $\boldsymbol{\chi_{min}}$. Fig. 6 a depicts 3-NNs of $\boldsymbol{x_i}$ adjoined by the blue lines with $\boldsymbol{x_i}$ instance. Secondly, SMOTE chooses arbitrarily an instance $\widehat{\boldsymbol{x}}_{\boldsymbol{i}}$ of the $\boldsymbol{k_{x_i}}$ that also belongs to $\boldsymbol{\chi}$ min to create a synthetic instance $\boldsymbol{x_{new}}$. Thirdly, a random number $\boldsymbol{\delta}$ is multiplied with the vector difference after a distance vector has been computed between $\boldsymbol{x_i}$ and $\widehat{\boldsymbol{x}}_{\boldsymbol{i}}$. The range of the $\boldsymbol{\delta}$ value is from 0 to 1, i.e., $\boldsymbol{\delta} \in [\boldsymbol{0}, \boldsymbol{1}]$. Finally, the feature vector of $\boldsymbol{x_{new}}$ is generated by following the formula:

$$x_{new} = x_i + (\widehat{x}_i - x_i) \times \delta \tag{3}$$

Here, the chosen instance $\widehat{\boldsymbol{x}}_{\boldsymbol{i}}$ belongs to $\boldsymbol{k_{x_i}} : \widehat{\boldsymbol{x}}_{\boldsymbol{i}} \in \boldsymbol{\chi_{min}}$. The feature vector is a synthetic instance, as shown in Fig. 6b, which lies between the segments connecting the line of $\boldsymbol{x_i}$ and the arbitrarily chosen $\widehat{\boldsymbol{x}}_{\boldsymbol{i}} \in \boldsymbol{k_{x_i}}$ based on Eq. 3, SMOTE replicates exact instances of the minority class, leading to more overgeneralisation. That can result in class overlapping because of disregarding neighbourhood heuristic rules. During resemblance, the minority classes rarely produce some redundant instances that do not contribute to the learning of those classes.

To overcome the drawbacks of SMOTE, Edited Nearest Neighbours (ENN) is applied to avoid not only enigmatic instances but also class overlapping. It neglects the synthetic instances, which are different from the 2 instances within 3-NNs, as depicted in **Fig. 7** a. The basic idea behind the ENN is that it identifies the neighbours of the targeted class instances using $\boldsymbol{k}$-Nearest Neighbours (NNs). Then, the ENN approach eliminates the instances if any or most of its neighbours belong to a different one. The flowchart of the SMOTE-ENN algorithm is described sequentially in **Fig. 7** b. It can also be said that SMOTE-ENN is a natural extension of SMOTE, whereas ENN functions as a data filter that removes noisy and ambiguous instances.

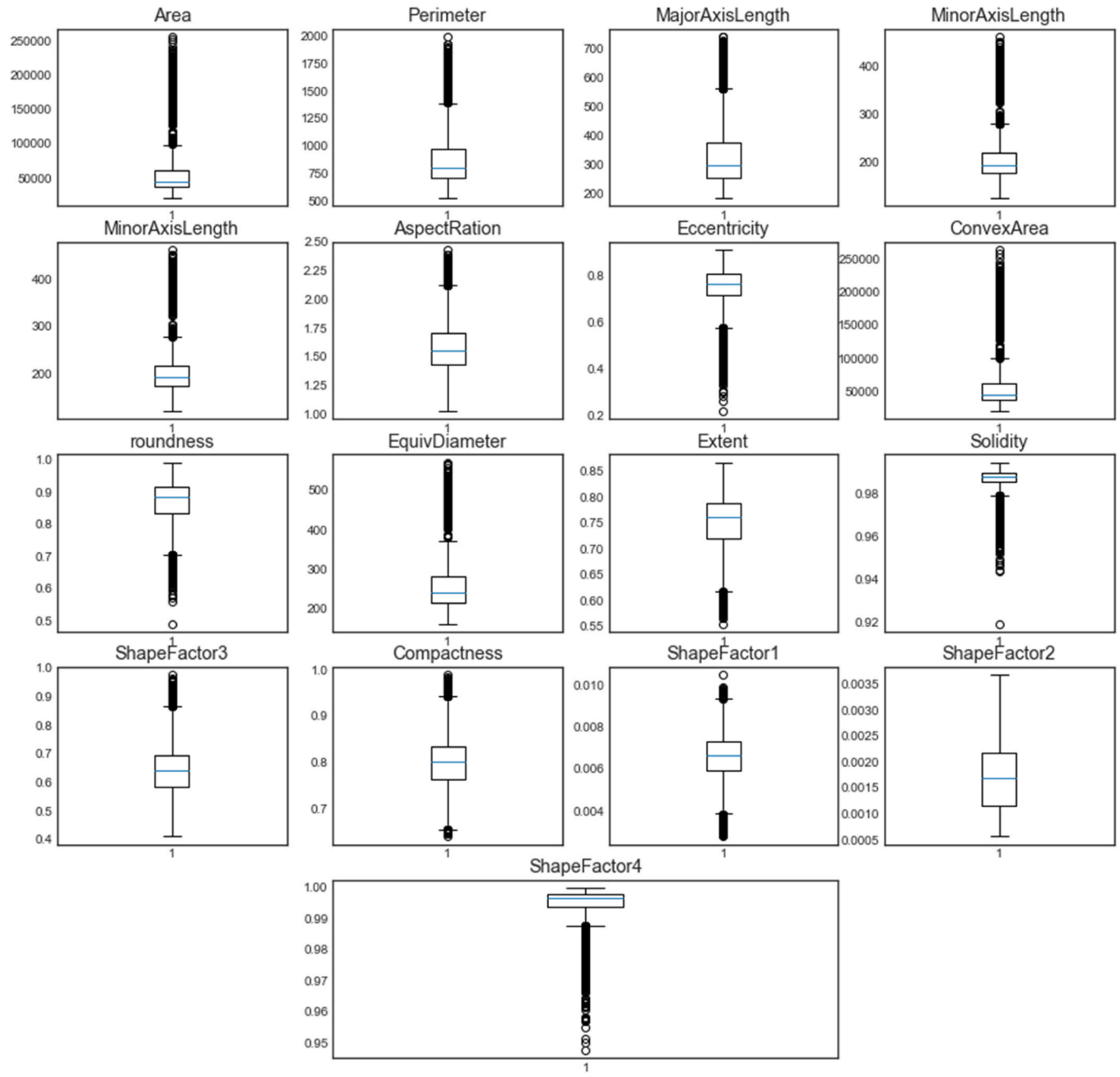


Fig. 3. The outlier's visualisation of dry beans using boxplot.

4. Machine learning technique

The Light Gradient Boosting Machine (LBM) is one of the most advanced tree-based ensemble learning tools for boosting predictions with incredible speed, interpretability and scalability. In 2016, Microsoft developed a model to handle large-scale datasets with numerous features and a wide range of subjects [38]. Its efficiency and flexibility come from employing discrete feature histograms to accelerate training speed and leaf-wise decision trees to reduce memory usage. It integrates Gradient-based One Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to address the drawback of the pre-sorted algorithm during the training of the Gradient Boosting Decision Tree (GBDT). The decision trees support learning the machine from the given space $\boldsymbol{x_{space}}$ to the gradient space $\boldsymbol{y}$ in GBDT. In GOSS, the roles of data instances vary on their gradients to calculate information gain; for example, the higher gradient instances get the priority to share more information gain. Then, GOSS removes the small gradient instances randomly, applying the under-sampling approach and considers the high gradient instances based on a pre-defined threshold to preserve the accuracy of information gained in computation.

For given a supervised training set $\boldsymbol{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \cdots, \boldsymbol{x_n}\}$ with xi vector in space, the negative gradients will be $\boldsymbol{Y} = \{\boldsymbol{y_1}, \boldsymbol{y_2}, \boldsymbol{y_3}, \cdots, \boldsymbol{y_N}\}$, which derive from the loss function in every iteration of gradient boosting corresponding to the model's outcome. Every node known as

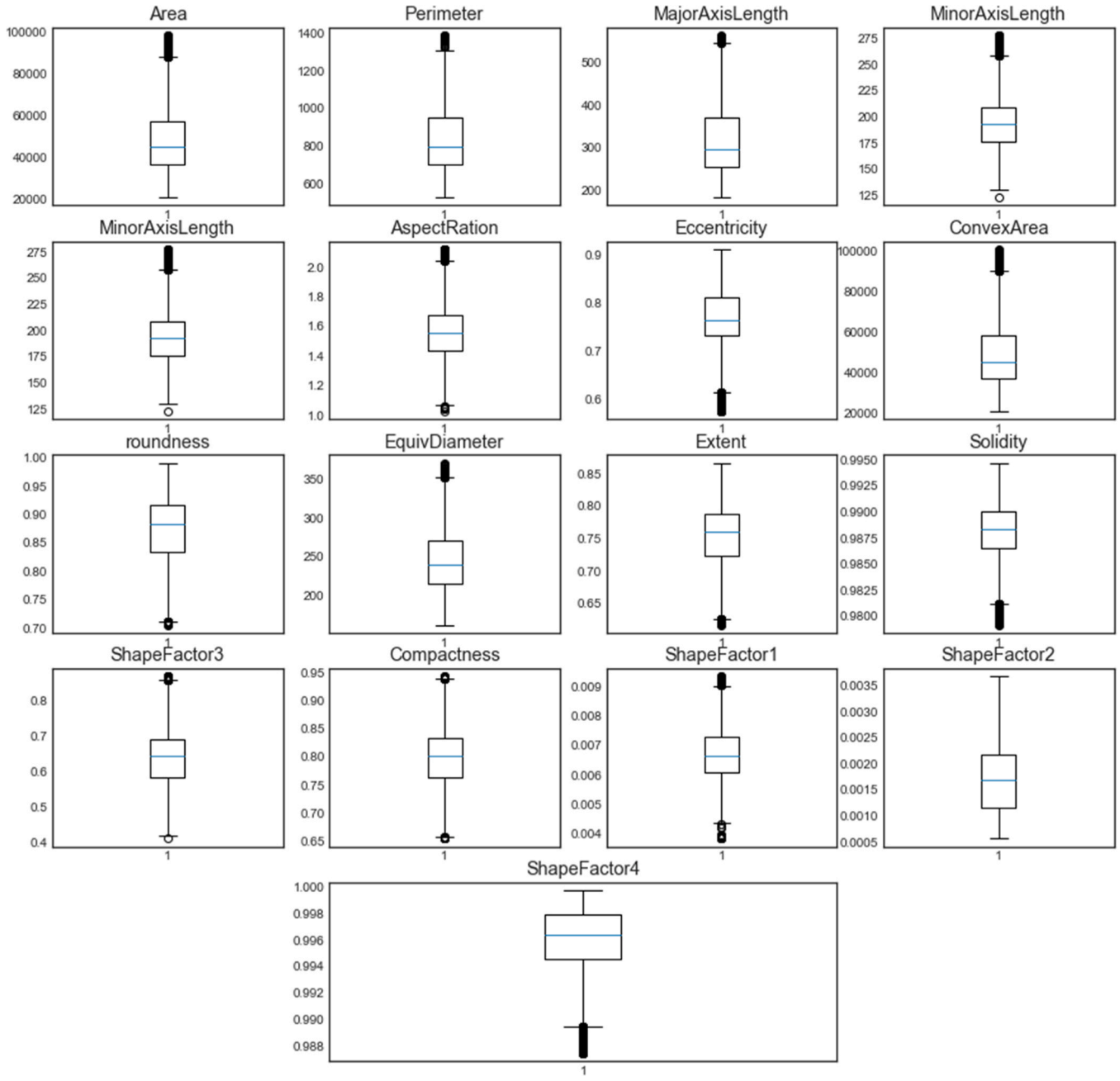


Fig. 4. The boxplot of geometric features after removing outliers.

the greatest information gain is split with the help of a decision tree of GOSS that is estimated by the variance $\boldsymbol{g}$. If $\boldsymbol{l}$ and $\boldsymbol{r}$ are the left and right nodes of the splitting feature $\boldsymbol{j}$ at point $\boldsymbol{d}$, then the variance gain will be estimated over the subset $\boldsymbol{p} \cup \boldsymbol{q}$ as follows:

$$\widehat{g}_f(h) = \frac{1}{n}\left[\left(\frac{\sum_{x_i \in p_l} y_i + (1-a)}{b\sum_{x_i \in q_l} y_i}\right) \Big/ n_l^f(h)\right] + \frac{1}{n}\left[\left(\sum_{x_i \in p_r} y_i + (1-a) \Big/ b\sum_{x_i \in q_r} y_i\right) \Big/ n_r^f(h)\right] \tag{4}$$

Herein, $\boldsymbol{p}$ denotes the subset of top $\boldsymbol{a} \times \boldsymbol{100}\%$ instances with greater gradient, i.e., $\boldsymbol{p_l} = \{\boldsymbol{x_i} \in \boldsymbol{p} : \boldsymbol{x_{if}} \leq \boldsymbol{h}\}; \boldsymbol{p_r} = \{\boldsymbol{x_i} \in \boldsymbol{p} : \boldsymbol{x_{if}} \leq \boldsymbol{h}\}$. Similarly, $\boldsymbol{q}$ is an arbitrary subset of $\boldsymbol{b} \times |\boldsymbol{p^c}|$ from the rest of $(\boldsymbol{1} - \boldsymbol{a}) \times \boldsymbol{100}\%$ instances of $\boldsymbol{p^c}$ with smaller gradients, i.e., $\boldsymbol{q_l} = \{\boldsymbol{x_i} \in \boldsymbol{q} : \boldsymbol{x_{if}} \leq \boldsymbol{h}\}$; $\boldsymbol{q_r} = \{\boldsymbol{x_i} \in \boldsymbol{q} : \boldsymbol{x_{if}} \leq \boldsymbol{h}\}$. Additionally, the total of $\boldsymbol{q}$ is normalised over the smaller gradients by the constant $(\boldsymbol{1} - \boldsymbol{a})/\boldsymbol{b}$. To find the split point, the computed gain $\widehat{\boldsymbol{g}}_{\boldsymbol{f}}(\boldsymbol{h})$ over the subset of a smaller instance, the subset is utilised rather than the accurate $\boldsymbol{g}_{\boldsymbol{f}}(\boldsymbol{h})$ across all the instances. Finally,

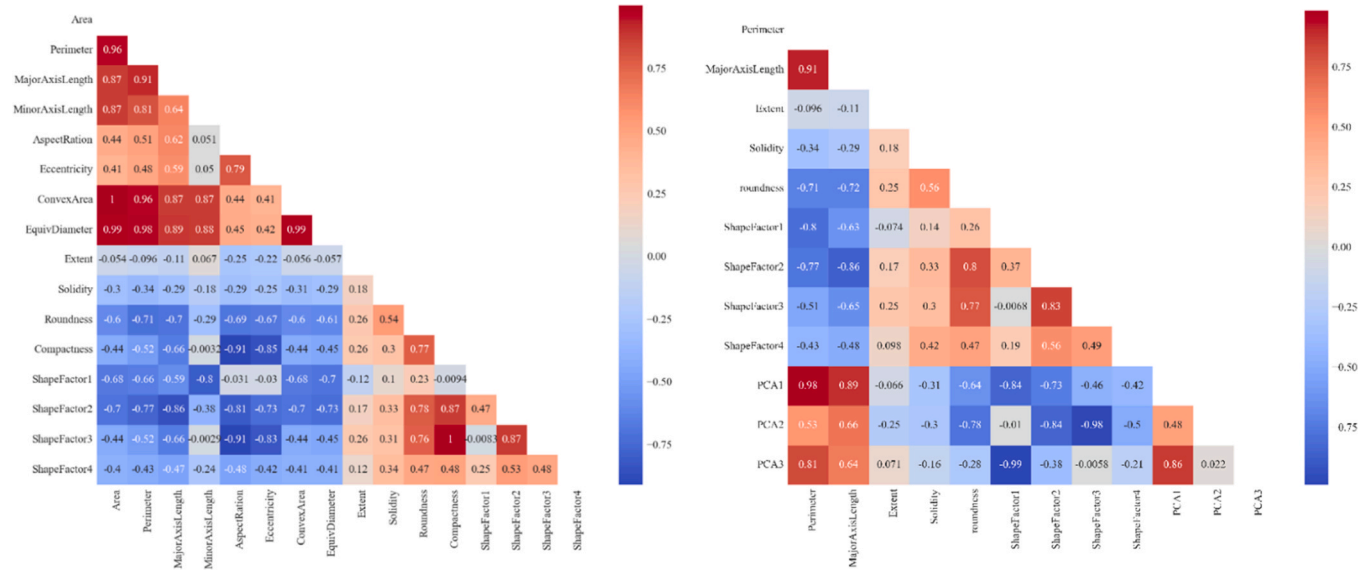


Fig. 5. (a) The correlation among the geometric features; (b) The selected features of dry beans after applying PCA.

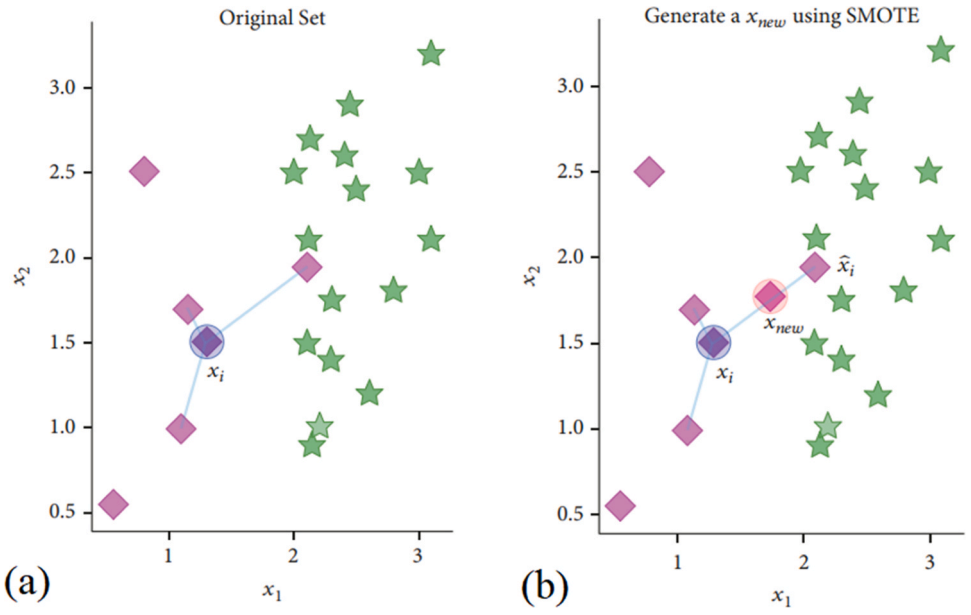


Fig. 6. (a) 3-NNs selection randomly for x_i; (b) The way of synthetic instance x_{new} generation.

the EFB technique merges the exclusive features into discrete bins. These bins employ building feature histograms that make the LBM model faster.

The LBM model defines logistic regression as the loss function $L(y, y_i)$ in [38]. During the training of the detector, the loss function works as an optimisation-based statistical model calibration because it chastises the inconsistency between true and predicted odds. It also compares the model-predicted outcomes with the target classes to quantify the relative uncertainty.

The Log loss function is formulated as follows:

$$L(y, y_i) = \frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}) + (1 - y_i)\log(1 - \hat{y})] \quad (5)$$

Where y_i and $\hat{y}$ represent the true class of the j-th instance and predicted probability.

4.1. Bayesian hyper-parameter Optimization

Most machine learning algorithms consist of several parameters, and even the classifier is the function of hyper-parameters. So, exploring the optimal combination of those parameters is more convenient. In this study, the concept of the Bayesian algorithm is chosen for hyper-parameter optimization, which is superior to manual, grid, random, and hunger game search techniques [39]. It keeps the records of prior evaluation metrics. Then, it utilizes them to build a probabilistic

Gaussian mode $\boldsymbol{P}(\boldsymbol{J})$ by dint of the objective function $\mathbf{J}(:)$. The weighted $\boldsymbol{f_1}$-score serves as the objective function $\mathbf{J}(:)$, evaluating the optimizer's performance after training the model with a specific hyper-parameter setting at each iteration. The optimizer selects the next set of hyperparameters by maximizing the weighted $\boldsymbol{f_1}$-score. The Bayesian optimization algorithm is given below.

Algorithm. Bayesian Hyper-parameter Optimization

1. Input: Initial training dataset, $\boldsymbol{D_{1:t-1}} = \{\boldsymbol{x_i}, \boldsymbol{y_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{t-1}}$ consists of $\boldsymbol{t-1}$ instances of object function $\boldsymbol{J}$
2. Output: $\{\boldsymbol{x_i}, \boldsymbol{y_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{t-1}}$
3. Construct a probabilistic model $\boldsymbol{P(J)}$ and acquisition function $\boldsymbol{a_{P(J)}}$
4. for $\boldsymbol{i}$ in range $(\boldsymbol{1}, \boldsymbol{t-1})$:
   a) Fit probabilistic model $\boldsymbol{P(J(x))}$ on training dataset $\boldsymbol{D_{1:t-1}}$ for the object function $\boldsymbol{J(x)}$
   b) Select $\boldsymbol{x_t}$ through the optimization of the acquisition function and across the function $\boldsymbol{J}$. $\boldsymbol{x_t} = \mathbf{arg\,max}\, \boldsymbol{a_{P(J)}}(\boldsymbol{x}; \boldsymbol{D_{t-1}})$
   c) Evaluate the objective function computationally to get $\boldsymbol{y_t}$ i.e., $\boldsymbol{y_t} = \boldsymbol{J(x_t)}$
   d) Augment the instances, $\boldsymbol{D_{1:t}} = \{\boldsymbol{D_{1:t-1}}, (\boldsymbol{x_t}, \boldsymbol{y_t})\}$
   e) Update the probabilistic model to prescribe the predicted objective model $\boldsymbol{J}$ i.e., $\boldsymbol{x_t^*} \leftarrow \mathbf{arg\,max}\{\boldsymbol{y_1}, \boldsymbol{y_2}, \cdots, \boldsymbol{y_t}\}$
5. end for

In hyper-parameter tuning, a single implementation is insufficient to account for model variability and adequately assess the selected parameters' reliability and stability. To address this concern, *K*-fold cross-validation averages performance across multiple train-test splits, providing a more reliable and stable estimate for each hyper-parameter setting. It ensures that the hyperparameters perform well across the entire dataset rather than being tailored to a specific partition. Additionally, this process helps stabilize performance estimates, making Bayesian optimization viable even when applied to unseen data that were not used during model training. The enlisted hyper-parameters in Table 3 are optimized with 12-fold cross-validation to determine its generalization ability and reduce the model's instability. The models consist of several hyperparameters, and the following hyperparameters are considered for optimization, which significantly influence the classification performances. Also, the default values are considered for the rest of the parameters.

4.2. Performance evaluation

Several evaluation metrics measure the efficiency, scalability and predictive accuracy of a classification model. The classification accuracy only computes the ratio of accurately predicted instances to all instances, which is insufficient to understand how robust a model is for multi-class classification problems. Among these metrics, average precision, recall, $\boldsymbol{F_1}$-score & accuracy are considered to measure the performance of the given multi-class problem [40]. These metrics are originated from a confusion matrix and mathematically defined as follows:

$$\text{Average Precision} = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FP_i} \tag{6}$$

$$\text{Average Recall} = \frac{1}{n}\sum_{i=1}^{n}\frac{TP_i}{TP_i + FN_i} \tag{7}$$

$$F_1 - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\frac{TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{9}$$

where $\boldsymbol{n}$ is the total number of classes; true positive ($\boldsymbol{TP}$) computes the number of beans that were positive and predicted as positive; true

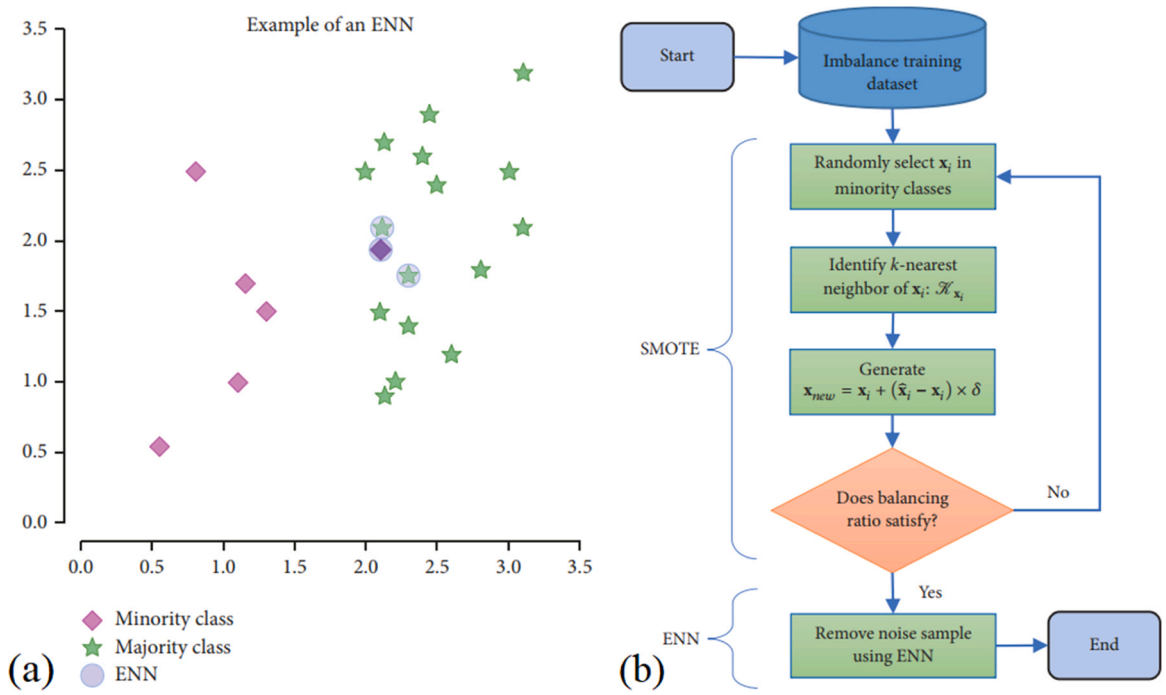


Fig. 7. (a) Neighbourhood cleaning rule based on ENN; (b) The flowchart of hybrid SMOTE-ENN resembling technique.

negative (***TN***) computes the number of beans that are negative and predicted as negative; false positive (***FP***) computes the number of beans that were negative and predicted as positive and false negative (***FN***) computes the number of beans that were positive but determined as negative for ***i*** class respectively.

Furthermore, ***K***-fold cross-validation is adopted to measure the fitness of prediction [7], which enhances the ability to generalize unknown seed varieties by conducting training and testing on a specific number of identical size sets several times.

4.3. Feature importance using SHAP analysis

The concept of SHAP analysis has been derived from game theory to illustrate the relative impact of each feature on the outcome of a target variable concerning the other features in a model [13]. The SHAP value has direction and magnitude, although the absolute value form of SHAP significance is used for model training. The linear function $\pounds(z')$ is formulated with the help of the additive attribute method.

$$f(z') = \mu_0 + \sum_{i=1}^{N} \mu_i z_i' \tag{10}$$

Where the total number of given features is denoted by N and $z' \in \{0,1\}^M$ is a coalition vector of the simplified features. The z_i' variables select the feature that is observed ($z_i' = 1$) or unknown ($z_i' = 0$). The feature attribution values ($\mu_i \in R$) are computed from the below equation.

$$\mu_i = \sum_{F_s \subseteq M/\{i\}} \frac{|F_s|!(n - |F_s| - 1)!}{n!} [\pounds_x(F_s \cup \{i\} - \pounds_x(F_x))] \tag{11}$$

Where F_s is the subset of features, M is the total input features.

5. Results and discussion

In this section, the dry beans dataset was split arbitrarily into training (80 %) and testing (20 %) sets to train the ML models. The experimental simulations are conducted on an Intel Core i5 laptop with 4 GB RAM and Python 3.7. Furthermore, the classifier's diagnostic ability is visualized and discussed using the Receiver Operating Characteristic curve (ROC) with Area under Curve (AUC), which plots the True Positive Rate (TPR) versus the False Positive Rate (FPR) for varying decision thresholds.

5.1. Optimal value exploration

A series of independent optimization trials were conducted using a Bayesian algorithm across different hyperparameter ranges to avoid convergence to local optima. Since different runs of the optimization might give different results depending on where the process starts or the randomness involved. The resulting accuracy and optimal values of hyperparameters obtained across these trials are explored with 12-fold cross-validation to support this process with the help of 30 iterations. Then, this study investigated how stable or consistent the outcomes were across different runs. Despite huge search ranges (e.g., learning rate, reg_lambda learning rate and subsample from [0.1, 10000], [0.001, 1500] and [0.001, 1.0]; max depth up to 5000 or n_estimators & num_leaves up to 10000), the optimization still consistently finds reasonable, non-extreme values.

If the optimization process were trapped in a local optimum, it would be expected that the resulting hyperparameter values would cluster irregularly or converge towards the boundaries of the defined search space (e.g., reaching maximum or minimum limits). However, the results presented do not exhibit such behaviour, indicating that the search effectively explored the space and avoided local optima.

Further, grid search (GSO), random search (RSO), Hunger game search (HGSO) and Bayesian optimization (BO) algorithms are adopted to evaluate the stability and robustness of these key hyperparameters and their optimal values. The optimization algorithm performance is listed in Table 4 using the pre-defined hyperparameters with optimal values of the LBM model. For the other techniques, such as GSO, RSO, and HGSO, the optimal hyperparameter values were estimated using a setup similar to that of BO, as shown in Table 5. The executing time is also computed in minutes (Mins.) to implement the program with computational cost. Although pre-defined parameters like criterion (gini), max features (log2) and metric (Minkowski) are run separately to

Table 3
Hyperparameter optimization including range and optimal values.

Classifiers (Clf)	No.of Parameters $z \in Z$	Hyperparameters	Range	Optimal Value
RF	4	n estimators	[100,400]	283
		max features		log2
		max depth	[1,20]	19
		criterion		gini
DT	6	max depth	[1,25]	22
		splitter		best
		criterion		gini
		min samples leaf	[1,15]	1
		min samples split	[2,20]	2
		random state	[1,50]	49
kNN	2	n neighbours	[3,10]	3
		metric		minkowski
SVM	3	kernel		rbf
		gamma	[1,1000]	1
		C	[0.001,21]	14.084612161396330
MLP	6	max iter	[100,500]	259
		alpha	[0.0001,0.11]	0.0217698117826773
		activation		tanh
		solver		lbfgs
		max fun	[1000,25000]	6432
		random state	[1,50]	42
CAT	5	depth	[1,16]	6
		learning rate	[0.01,1]	0.1899999976158142
		n estimators	[100,800]	500
		l2 leaf reg	[1,9]	3
		rsm	[0.1,1]	0.9999878675433323
GBC	5	max depth	[1,15]	12
		learning rate	[0.01,1]	0.1072375031768350
		n estimators	[100,400]	120
		max features		log2
		subsample	[0.1,1]	0.9082556675433323
XGB	5	max depth	[1,15]	14
		learning rate	[0.1,1]	0.1170370031768306
		n estimators	[1,1000]	117
		subsample	[0,1]	0.5624877067927799
		reg lambda	[0,1]	0.9082556675433323
LR	4	max iter	[10, 500]	273
		C	[1, 500]	306.74327276709280
		l1 ratio	[0.01,1]	0.9019875584709440
		tol	[0.0001,1]	0.3877247381193835
LBM	7	boosting type		gbdt
		max depth	[1,20]	5
		learning rate	[0.1, 1]	0.5141185668939924
		reg_lambda	[0.1, 1]	0.3892109603412833
		subsample	[0.1, 1]	0.5019007442352383
		n_estimators	[100, 400]	110
		num_leaves	[20, 100]	50

reduce the computational cost, the executing time is high. The consistency across all metrics reflects BO's robustness and stability, even in high-dimensional or complex feature spaces. Also, the highest cross-validation score suggests it generalizes unseen data better than others.

These optimal values consistently converged and remained stable across the intervals. It shows that the Bayesian optimization process was not trapped in a poor local optimum, even in more complex spaces, as evidenced by the results from Table 5. The BO technique outperformed other tuning strategies, such as Grid Search (99.53 %), Hunger Game Search (99.48 %), and Random Search (99.46 %). The superior performance of the BO technique, both in trials and comparative results, validates its ability to efficiently explore the hyperparameters and avoid local optima, leading to better generalization and higher predictive accuracy.

5.2. Model's sensitivity analysis

The SMOTE-ENN resampling technique exhibits high sensitivity to the choice of the k parameter, as it simultaneously governs the synthetic instance generation in SMOTE and the noise filtering strength in ENN, directly impacting model performance. A sensitivity analysis was conducted to investigate this effect by varying the nearest neighbours ($\boldsymbol{k}$) used in both techniques. The corresponding table evaluates how different combinations of k values in SMOTE and ENN impact the model's performance.

The 3D plot illustrates the variation in misclassification rates as a function of the number of neighbours (k) in both ENN and SMOTE methods across different model configurations on the dry bean dataset. It is clear from Fig. 8a that performance remains consistently high (above 0.99) across almost all combinations. This suggests that the model is not overly sensitive to specific k values and performs reliably within a reasonable range ($k = 3to7$). The overall performance tends to remain high when both SMOTE and ENN are in the range of 4–7, though using very low values (e.g., $k = 3$) occasionally leads to minor drops in performance. The highest performance (0.9959) was achieved when $k = 5$ in SMOTE and $k = 3$ in ENN. This suggests that generating synthetic instances based on 5 nearest neighbours while filtering noise using 3 neighbours provides consistency between sample diversity and noise reduction. These findings indicate that the choice of k significantly influences model performance and should be carefully tuned according to the specific characteristics of the dataset. In the case of dry bean classification, the LBM model achieved the best performance, attaining a misclassification rate of only 0.4 % with optimally tuned hyperparameter settings.

5.3. Influence of *K*-fold Cross-validation in Bayesian optimization

In Bayesian hyperparameter optimization, *K*-fold cross-validation reduces variance. It provides a more accurate estimate of model performance, helping the optimization process find the best hyperparameters more effectively. Fig. 8b presents the performance metrics of the SMOTE-ENN resampling strategy, evaluated in terms of precision, recall, F_1-score, accuracy (%), and cross-validation score across different *K*-fold cross-validation settings (*K* ranging from 5 to 14). The model consistently demonstrates exceptionally high performance, with precision, recall, and F_1-scores ranging from approximately 0.994 to 0.996 and accuracy varying between 99.41 % and 99.59 %. These results indicate that the model is highly accurate and reliable with very low misclassification rates. Higher *K* values ($K = 13, 12, 11, 8, 7$) resulted in slightly improved accuracy and cross-validation scores compared to lower *K* values. However, excellent performance was maintained across all settings. This trend reveals that the higher *K*-fold values can give a better generalization at the cost of longer computation time. Notably, all cross-validation scores exceeded 0.985, which confirms excellent generalization ability — the model performs consistently well on unseen data. The highest performance was achieved with $K = 12$, yielding an accuracy of 99.59 % and a cross-validation score of 0.9886, suggesting that $K = 12$ folds might be optimal among the tested values. Additionally, the precision, recall, and F_1-score are almost identical across all folds, showing that the model is well-balanced (i.e., neither overpredicts positives nor negatives).

5.4. Resampling performance

In real life, most classification problems are not uniformly distributed into the class variants, i.e., they are imbalanced. To address this issue, the discriminatory power of Synthetic Minority Oversampling Technique-Edited Nearest Neighbours (SMOTE-ENN) is compared to the customized NearMiss, RUS, Random Over Sampler (ROS), Adaptive Synthetic (ADASYN) sampling approach, Synthetic Minority Oversampling Technique (SMOTE), Synthetic Minority Oversampling Technique-Borderline (SMOTE-Borderline), Synthetic Minority Oversampling Technique-Support Vector Machine (SMOTESVM), and Synthetic Minority Oversampling Technique-Tomek (SMOTETomek) with similar settings of LBM classifier.

It is clear from Table 6 that the oversampling techniques improve the success rate of minority classes compared to undersampling. The NearMiss and RUS perform equally to "None" because they randomly eliminate instances from their targeted classes. In contrast, the accuracy of the ADASYN and SMOTE methods will converse in similar scores. On average, the ROS outperforms SMOTE by 0.61 % and ADASYN by 0.64 % in the $\boldsymbol{F_1}$-score with random replication of minority samples. However, the prior SMOTE method adds noisy and irrelevant instances during the oversampling of the minority class that overlaps overlapping classes and makes inaccurate predictions. It occurs not to focus on the relevant or quality synthetic instance and not consequently to officiate the underlying distribution of the minority class. The hybrid algorithms' precision, recall and $\boldsymbol{F_1}$-score are significantly improved compared to the SMOTE algorithm.

The hybrid approaches show that SMOTE-Tomek, SMOTE-Borderline, and SMOTE-SVM perform equally well. At the same time, SMOTE-Borderline and SMOTESVM reduce performance by 0.02 %, and SMOTE-Tomek increases performance by 0.34 % compared to SMOTE-Borderline. The results converge toward a consistent accuracy level, as shown in **Fig. 9** a; however, ENN significantly improves the generalization performance of SMOTE with the highest classification accuracy of 99.59 %. Regarding the AUC, a few resembling approaches like "None, NearMiss, RUS, ADASYN and SMOTE" acquire low accuracy but show high AUC due to bias toward positive classes. It is evident from Table 6 that SMOTE-ENN improves performance by approximately 3 % with low computational cost, as the technique effectively avoids noise and ambiguous instances in the synthetic data by combining oversampling with intelligent data cleaning. Additionally, this hybrid resampling technique reduces model complexity and enhances the model's ability to generalize well to unseen data.

5.5. Classification performance

The model's performance measures thoroughly against a set of cutting-edge classifiers, for instance, DT, LR, RF, gradient boosting (GBC), XGB, MLP, categorical gradient boosting (CAT), SVM, KNN, and LBM, as listed in Table 7. The $\boldsymbol{F_1}$-scores have overlapped in most classifiers that allude to a balanced evaluation of the model's performance, especially in scenarios with imbalanced data.

At first glance, the performance of the SVM and KNN classifiers may appear similar in Table 7. However, the SVM classifier performs significantly better than KNN in terms of ROC analysis, as shown in Fig. 9b. The LBM model outperforms all other classifiers with the highest accuracy of 99.59 %, along with the top F_1-score (0.9957), indicating an excellent balance between precision and recall. It slightly edges out SVM and KNN, achieving 99.57 % accuracy. However, LBM maintains a marginally better F_1-score, demonstrating more consistent performance in Table 7. Meanwhile, ensemble methods like XGB, GBC, and RF also perform strongly, with accuracies above 99.2 %, but not as good as LBM's results. In contrast, the DT and LR models show considerably lower accuracies of 98.01 % and 98.32 %, respectively, indicating that they may not capture complex patterns in the data as effectively as boosting or deep learning-based models like MLP (99.50 %).

During the evaluation, the Horoz variety always shows the lowest classification performance for all models. It is misguided by the Sira variety because of overlapping in the feature space, i.e., flatness and roundness. Also, most of the classifiers are higher than 99.28 % in accuracy. However, DT and LR have the lowest accuracy, around 98 %, in all the metrics. The improved LBM model performs best due to its ability to mitigate overfitting and effectively handle class imbalance by assigning weights to minority and majority classes. Combined with a hybrid resampling technique, this approach further enhances the model's robustness. Notably, tuning influential hyperparameters using Bayesian optimization improves the performance of LBM classifier by 0.13 %, demonstrating its effectiveness. Furthermore, estimating a mean AUC for the ROC curve provides an additional layer of statistical validity. Together, these techniques contributed to a more accurate, efficient and interpretable solution for multi-class classification in imbalanced datasets.

Table 4
The optimal values of the optimization techniques, including hyperparameters and their intervals.

Methods	List of hyperparameters with their ranges						
	boosting type	max depth	learning rate	reg_lambda	subsample	n_estimators	num_leaves
	['gbdt', 'dart']	[1,20]	[0.1, 1]	[0.1, 1]	[0.1, 1]	[100, 400]	[20, 100]
Bayesian optimization	gbdt	5	0.5141185668939924	0.3892109603412833	0.5019007442352383	110	50
Hunger game search	gbdt	18	0.17221455	0.59364584	0.20916457	159	59
Grid search	gbdt	6	0.19	0.35	0.2	120	40
Random search	gbdt	7	0.1	0.1	0.1	279	29

Table 5
Comparison of performance with optimization techniques.

Optimization Technique	Avg. Precision	Avg. Recall	F_1-score	Accuracy (%)	CV score (%)	Time (Mins.)
GSO	0.9950	0.9950	0.9950	99.53	98.77	72.77
RSO	0.9943	0.9943	0.9943	99.46	98.77	33.49
BO	0.9957	0.9956	0.9957	99.59	98.86	22.49
HGSO	0.9945	0.9946	0.9946	99.48	98.83	32.39

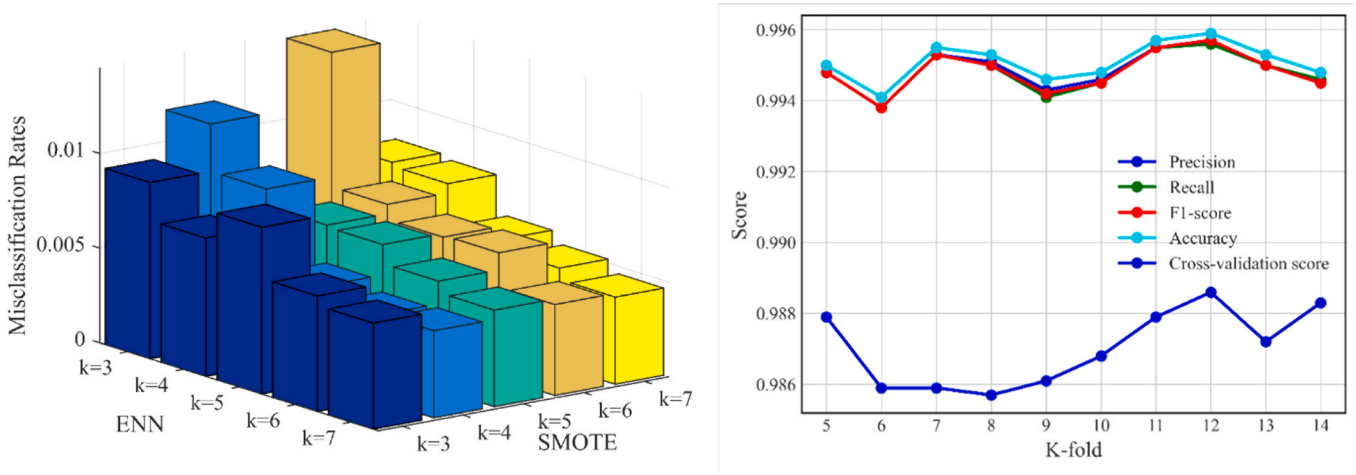


Fig. 8. (a) The model's sensitivity to the value of k in both SMOTE and ENN; (b) Illustration of *K*-Fold cross-validation for reliable model performance estimation.

5.6. Summary of seed classification

The classification of dry bean varieties is performed using an improved LBM classifier optimized via Bayesian methods and enhanced by the SMOTE-ENN resampling technique. Table 8 presents the confusion matrix for the classification of dry bean varieties, displaying both the raw classification counts and their corresponding percentage values. Correct classifications are located along the matrix diagonal, while off-diagonal elements represent misclassifications. The classifier demonstrates high overall accuracy, correctly identifying 4464 out of 4482 dry beans, yielding an accuracy of 99.59 % and an error rate of 0.41 %. Notably, the Bombay class achieved 100 % accuracy, indicating perfect model performance for that variety, likely due to its distinctive features representation in the training data. The Seker, Barbunya, Cali, Dermason, Sira, and Horoz classes followed closely, with accuracy rates of 99.70 %, 99.70 %, 99.55 %, 99.46 %, 98.74 % and 99.55 %, respectively. These high performances indicate that the model effectively distinguishes these classes, although minor misclassifications occurred—for example, 8 Dermason beans were classified as one Seker, one Horoz, and six Sira beans. A few bean varieties—except for Bombay—were confused with other varieties due to inter-class similarities and overlapping morphological characteristics. Overall, the model demonstrates excellent reliability in dry bean classification, and its strong performance underscores its suitability for high-accuracy agricultural sorting and automated quality control tasks.

Table 6
Comparison performance of resampling algorithms.

Resample Algorithms	Avg. Precision	Avg. Recall	F_1-score	Accuracy (%)
None	0.9362	0.9339	0.9350	92.17
NearMiss	0.9304	0.9290	0.9289	92.88
RUS	0.9268	0.9261	0.9258	92.61
ROS	0.9641	0.9641	0.9640	96.41
ADASYN	0.9577	0.9578	0.9576	95.81
SMOTE	0.9581	0.9579	0.9579	95.79
SMOTE-Borderline	0.9617	0.9613	0.9612	96.13
SMOTE-SVM	0.9610	0.9611	0.9610	96.11
SMOTE-Tomek	0.9645	0.9643	0.9643	96.47
SMOTE-ENN	0.9957	0.9956	0.9957	99.59

5.7. Performance of SMOTE-ENN technique

This study employs a learning curve to illustrate the proposed model's performance in the training data perspective. The boldface curves refer to the mean score values, and the light-shaded region surrounding the curves denotes the range of its standard deviation. The red and blue curves for training and testing scores depict the model's statistical performance using 12-fold cross-validation.

Though the training score is consistently high along with the iterations irrespective of the training set's size, a sign of proper fit is visualized in Fig. 10 a with the increased testing score. At a time, its performance has reached a fixed point that is not good enough to deal with realistic seeds. This implies that the model's performance can be improved by tuning hyper-parameters, selecting features/engineering, or collecting more training seeds. The convergence of learning curves in Fig. 10 b at a satisfactory score indicates that the LBM model is neither overfitting nor underfitting. The training score is always high through the iterations and training seeds. On the contrary, the training seed size remains constant while the test score increases. Indeed, it rises until it hits a plateau, which indicates that it may no longer be convenient to add more seeds for the model's training as the ability of generalization will no longer improve. The learning curves demonstrate that a robust, complex model is built to capture all the complexity in the seeds.

5.8. Feature importance using SHAP analysis

The importance of each feature is illustrated in the summary plot, which includes the distribution of each class. The priority of each feature is rearranged from top to bottom in descending order. Each feature's colour distribution represents each class's average absolute SHAP values to allow comparison by class. For example, "Perimeter" is the most important feature among the geometric features, and the contribution of SHAP value for the "Dermason" variety is high over the other seed varieties. Fig. 11 shows simultaneously that the importance of "Solidity",

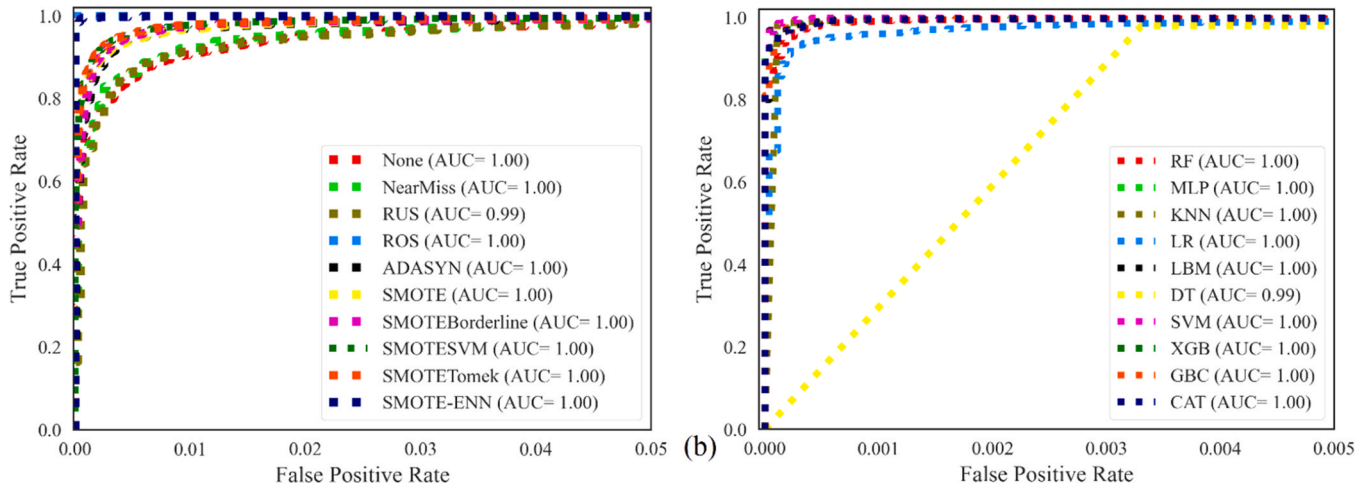


Fig. 9. (a) Effects of resampling algorithms; (b) Performance of improved LGB classifier.

Table 7
Classification performances with a set of cutting-edge classifiers.

Name of Classifiers	Avg. Precision	Avg. Recall	F_1-score	Accuracy (%)
DT	0.9795	0.9794	0.9794	98.01
LR	0.9828	0.9830	0.9829	98.32
RF	0.9925	0.9924	0.9925	99.28
GBC	0.9936	0.9936	0.9936	99.39
XGB	0.9938	0.9939	0.9938	99.41
MLP	0.9949	0.9949	0.9949	99.50
CAT	0.9948	0.9948	0.9948	99.50
SVM	0.9957	0.9955	0.9956	99.57
KNN	0.9953	0.9957	0.9955	99.57
LBM	0.9957	0.9956	0.9957	99.59

"PCA1", and "PCA2" are the least important features to classify uniform seed classification, as the SHAP values of those features are significantly lower than others.

5.9. Comparative study

A detailed comparison of recent studies is provided in Table 9, which includes various resampling and classification techniques to address the challenge of data imbalance. The accuracy of our results should not be directly compared with those of other studies. In [41], Coronel et al. have taken into account 7 types of morphological shapes as discriminative features of the beans. The experimental dataset comprises 3820 dry beans from 3 varieties, and the accuracy of SVM was 93.12 % on 764 beans with overfitting. However, our approach substantially enhances classification performance, consistent with prior research on imbalanced multi-class classification problems. For instance, studies without explicit resampling, such as those by Lopes et al. [42], Prasad et al. [43], and Souza et al. [43], reported accuracies of 93.18 % for MLP, 95.00 % for ANN, and 98.18 % for the neuro-fuzzy network, respectively. In contrast, Koeshardianto et al. [44], Mucuacua et al. [21], and Nayak et al. [45] slightly improved performance by applying SMOTE resampling with ensemble and kernel-based classifiers. Krishnan et al. [46] used RUS with CAT and achieved 95.35 % accuracy, while Khan et al.

Table 8
Summary of seed classification using confusion matrix.

		Predicted Class							
		Seker	Barbunya	Bombay	Cali	Horoz	Sira	Dermason	Total
True Class	Seker	663 (14.79 %)	0	0	0	0	1 (0.02 %)	1 (0.02 %)	665 (99.70 %) (0.30 %)
	Barbunya	0	659 (14.70 %)	0	2 (0.04 %)	0	0	0	661 (99.70 %) (0.30 %)
	Bombay	0	0	709 (15.82 %)	0	0	0	0	709 (100 %) (0 %)
	Cali	0	0	0	662 (14.77 %)	1 (0.02 %)	0	0	663 (99.85 %) (0.15 %)
	Horoz	0	0	0	1 (0.02 %)	665 (14.84 %)	0	1 (0.02 %)	667 (99.70 %) (0.30 %)
	Sira	0	1 (0.02 %)	0	0	1 (0.02 %)	550 (12.27 %)	1 (0.02 %)	553 (99.46 %) (0.54 %)
	Dermason	1 (0.02 %)	0	0	0	1 (0.02 %)	6 (0.13 %)	556 (12.41 %)	564 (98.58 %) (1.42 %)
Total		664	660	709	665	668	557	559	4482
Accurate Rate		(99.85 %)	(99.85 %)	(100 %)	(99.55 %)	(99.55 %)	(98.74 %)	(99.46 %)	(99.59 %)
Error Rate		(0.15 %)	(0.15 %)	(0 %)	(0.45 %)	(0.45 %)	(0.26 %)	(0.54 %)	(0.41 %)

[10] reported 95.40 % using ADASYN and XGB techniques.

Lee et al. [49] applied natural extensions of SMOTE resampling to address imbalanced data, utilizing BLSMOTE and ***k***-means clustering algorithms. The classification accuracy of BLSMOTE + ***k***-means + SVM on the dry bean dataset is 97.54 %, which is better than 96.98 % of ***k***-means + SVM, 93.03 % of BLSMOTE +SVM and 93.75 % of only SVM. The SMOTE resampling variant introduced by Dejene et al. [47] achieved 93.03 % accuracy with a soft voting classifier. However, it was not as effective as the approach by Lee et al. Although these methods [43,45,49] are effective, they lack the synergistic benefits of hybrid resampling and automated hyper-parameter tuning. More recent approaches, such as the deep-learning-based GA-SMOTE framework by Alex et al. [33], have achieved high accuracy but at the cost of greater computational complexity. Compared to these, our proposed SMOTE-ENN + Bayesian-optimized LBM framework achieves a superior accuracy of 99.59 % with fewer computational requirements and better generalization. Moreover, the improved LBM model has the lowest likelihood of being biased in favour of the majority class and is less likely to be distracted by various noise implications when fusing geometric features. Furthermore, studies such as Pepsi et al. [32] illustrate the growing importance of hybrid optimization in classification, a trend that our work supports through empirical evidence and comparative analysis. The superior performance on the experimental dataset demonstrates that the proposed framework balances complexity, interpretability, and high accuracy, setting a new benchmark for robust multi-class classification in imbalanced datasets.

5.10. Advantages, disadvantages, and future studies

Research on dry beans has practical implications in agriculture, particularly in quality control, sorting, and breeding. Accurate classification can lead to better sorting of beans based on quality, size, and other characteristics, improving overall product quality. The proposed framework can classify beans genetically, which varieties are drought-tolerant and disease-resistant under certain climates. This allows farmers to plant the right variety of beans for the expected climate, increasing yield and reducing crop failure.

The improved LBM model was evaluated using a benchmark and one of the largest multi-class datasets, considering only geometrical features such as dimensional and shape features, which carry no information about the bean colour. Dry beans are sensitive to environmental factors like soil quality, weather, and irrigation, limiting the applicability of research findings across different regions and climates. The feature selection strategy performs better for this dataset to overcome the inter-seed ambiguities but may not work well for other datasets. The decision was taken to drop irrelevant features, which are highly correlated and echoed, to avoid the model's overfitting. The model was established and optimized for dry bean classification. Its architecture is inherently flexible due to its reliance on geometric features. The addressed issues demonstrate the model's adaptability, especially when dealing with small variations in appearance between bean varieties, where other features might fail to capture the subtle differences.

Besides the raw geometric features in this study, incorporating colour, texture, and 3D properties can accelerate dry bean classification by capturing visual and structural details beyond basic shape and size. Colour helps distinguish between visually similar beans that may have overlapping geometric characteristics but differ in pigmentation. Meanwhile, texture analysis captures surface patterns and irregularities not evident from shape alone. Additionally, the 3D property, like the suture axis, adds depth and volumetric information, enabling more accurate modelling of bean size, curvature, and surface structure. This can accelerate classification performance by capturing physical traits absent in 2D feature analysis. In industry, seeds move so quickly through the orifice of classification machines that 3D analysis becomes more difficult and time-consuming.

In addition to achieving satisfactory accuracy for real-life applications, the hybridization of upcoming algorithms, along with ML and DL techniques, to automatically learn higher levels of abstraction from raw data may be employed to accelerate the model further. After all, geometric features are essential for reliable multi-class classification. In contrast, colour and texture features are often insufficient or inconsistent due to lighting, processing, and natural overlap variations. Further studies could explore the generalization of this approach to other agricultural datasets, such as dry beans, wheat, maize, and sunflower seeds. These studies assess its applicability and effectiveness across different crop types and environmental conditions.

6. Conclusions

This work uses data mining approaches to conduct an exploratory analysis of the dried beans. The outcomes of the experiment imply that applying data mining approaches such as feature selection, dataset balance, outlier detection, and robust machine learning algorithms can enhance the quality of seed classification. Applying the LBM model with hyper-parameter optimization and SMOTE-ENN adjustment yielded the highest accuracy of 99.59 %. The classification performances were 92.94 % for the original imbalanced dataset, 92.36 % using outlier detection, 95.70 % using SMOTE except for feature selection, 95.79 % using feature selection & SMOTE, and 99.59 % using feature selection & SMOTE-ENN hybrid techniques. The overall accuracy increased

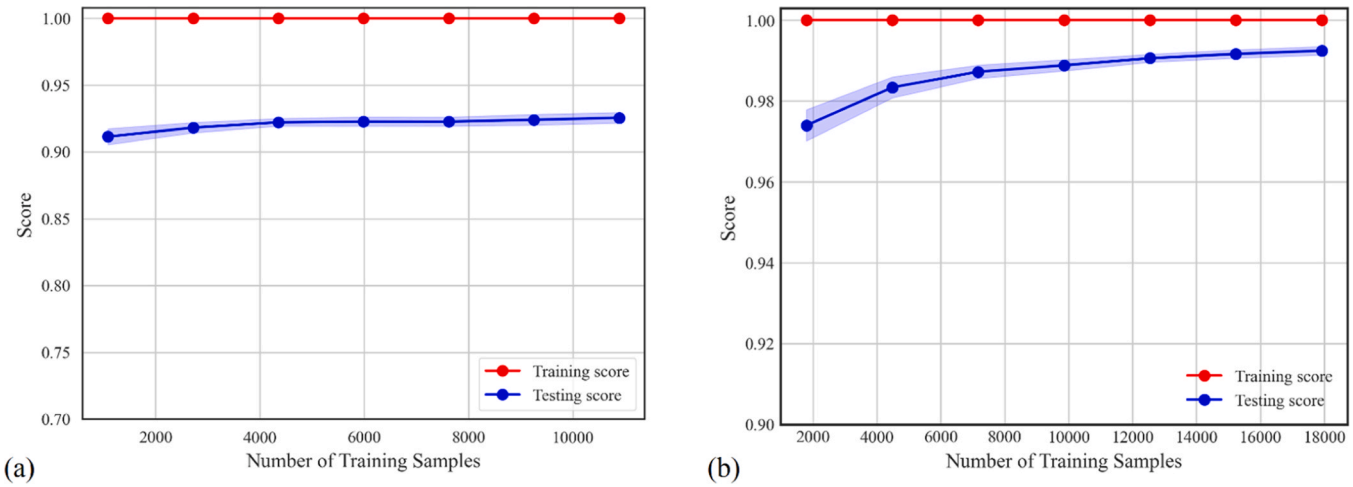


Fig. 10. (a) The learning curve without SMOTE-ENN technique; (b) The learning curve with SMOTE-ENN technique.

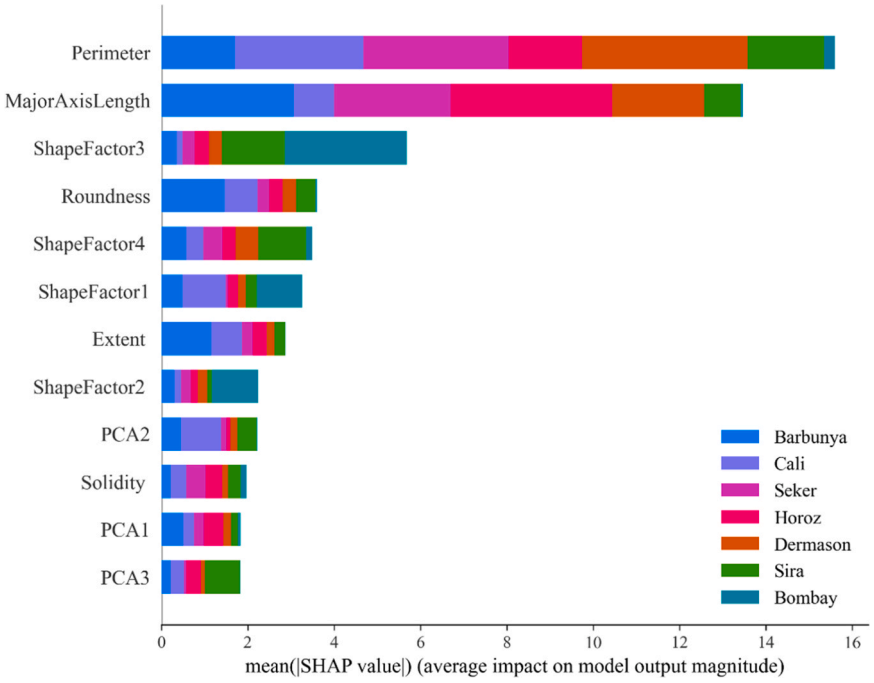


Fig. 11. Feature importance plot using SHAP.

Table 9
Comparison with related works on Dry bean seeds.

Name of Authors	Resampling	Selected Classifier	Accuracy (%)
Dejene et al. [47]	SMOTE + Tomek	Soft Voting Classifier	92.65
Lopes et al. [42]	-	Multi-layer Perceptron	93.18
Prasad et al. [48]	-	Artificial Neural Network	95.00
Koeshardianto et al. [44]	SMOTE	Stochastic Gradient Boosting Machine	95.38
Krishnan et al. [46]	RUS	Categorical Gradient Boosting	95.35
Khan et al. [10]	ADASYN	Extreme Gradient Boosting	95.40
Macuacua et al. [21]	SMOTE	**k**-Nearest Neighbour	96.00
Nayak et al. [45]	SMOTE	Extreme Gradient Boosting	97.32
Lee et al. [49]	BLSMOTE + ***k***-means	Support Vector Machine	97.54
Souza et al. [43]	-	Neuro-fuzzy Network	98.18
Proposed	SMOTE-ENN	Light Gradient Boosting Machine	99.59

significantly by about 7 % when the data mining strategies were applied sequentially. The comparisons with earlier approaches guarantee that the proposed framework automatically weighs up the variations of features' relevance with a view to quickly and robustly classifying dry bean varieties for evaluation. The improved LMB model proves the feasibility of automatically classifying dry beans into genetic variations of different planting areas in Turkey. Thus, our proposed method could be integrated into real-time seed sorting systems, benefiting small-scale farmers and food industries by ensuring faster and more accurate classification. Additionally, these sorting systems help identify bean varieties that meet basic standards for planting and marketing, thereby promoting both producer and customer satisfaction.

CRediT authorship contribution statement

Shahina Akter: Visualization, Validation, Methodology, Investigation. **Chalak Qazani Mohamad Reza:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **B.M. Jewel Rana:** Visualization, Validation, Software, Resources. **Arnab Mukherjee:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Houshyar Asadi:** Writing – review & editing, Supervision, Resources, Project administration. **Lasker Ershad Ali:** Visualization, Validation, Methodology, Investigation. **Md. Salauddin Khan:** Visualization, Validation, Methodology, Investigation. **Amirhossein Mohajerzadeh:** Visualization, Validation, Methodology, Investigation. **Nusrat Jahan Sathi:** Visualization, Validation, Methodology, Investigation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] M. Altalhan, A. Algarni, M.T.-H. Alouane, Imbalanced data problem in machine learning: a review, IEEE Access (2025).
[2] S. Rezvani, X. Wang, A broad review on class imbalance learning techniques, Appl. Soft Comput. 143 (2023) 110415.
[3] T. Miftahushudur, et al., A survey of methods for addressing imbalance data problems in agriculture applications, Remote Sens. 17 (3) (2025) 454.
[4] Z. Ren, et al., A systematic review on imbalanced learning methods in intelligent fault diagnosis, IEEE Trans. Instrum. Meas. 72 (2023) 1–35.
[5] S. Goswami, A.K. Singh, A literature survey on various aspect of class imbalance problem in data mining, Multimed. Tools Appl. 83 (27) (2024) 70025–70050.

[6] J.Y. Yu, et al., Evaluation of conventional and quantum computing for predicting mortality based on small early-onset colorectal cancer data, Appl. Soft Comput. 162 (2024) 111781.
[7] M. Koklu, I.A. Ozkan, Multiclass classification of dry beans using computer vision and machine learning techniques, Comput. Electron. Agric. 174 (2020) 105507.
[8] Y.S. Taspinar, et al., Computer vision classification of dry beans (Phaseolus vulgaris L.) based on deep transfer learning techniques, Eur. Food Res. Technol. 248 (11) (2022) 2707–2725.
[9] M. Dogan, et al., Dry bean cultivars classification using deep cnn features and salp swarm algorithm based extreme learning machine, Comput. Electron. Agric. 204 (2023) 107575.
[10] M.S. Khan, et al., Comparison of multiclass classification techniques using dry bean dataset, Int. J. Cogn. Comput. Eng. 4 (2023) 6–20.
[11] Z. Liu, et al., A survey of imbalanced learning on graphs: problems, techniques, and future directions, IEEE Trans. Knowl. Data Eng. (2025).
[12] W. Chen, et al., A survey on imbalanced learning: latest research, applications and future directions, Artif. Intell. Rev. 57 (6) (2024) 137.
[13] K. Debjit, et al., An improved machine-learning approach for COVID-19 prediction using Harris Hawks optimization and feature analysis using SHAP, Diagnostics 12 (5) (2022) 1023.
[14] M.S. Islam, et al., HGSORF: Henry Gas Solubility Optimization-based Random Forest for C-Section prediction and XAI-based cause analysis, Comput. Biol. Med. 147 (2022) 105671.
[15] R. Subathra, V. Sumathy, An offbeat bolstered swarm integrated ensemble learning (BSEL) model for heart disease diagnosis and classification, Appl. Soft Comput. 154 (2024) 111273.
[16] T.M. Alam, et al., An investigation of credit card default prediction in the imbalanced datasets, Ieee Access 8 (2020) 201173–201198.
[17] S.F. Fahim, et al., Classification of dry beans into genetic varieties using deep learning-based Convolutional Neural Networks (CNNs). International Conference on Cognitive Computing and Cyber Physical Systems, Springer, 2023.
[18] I.Y. Hafez, et al., A systematic review of AI-enhanced techniques in credit card fraud detection, J. Big Data 12 (1) (2025) 6.
[19] M. Han, H. Guo, W. Wang, A new data complexity measure for multi-class imbalanced classification tasks, Pattern Recognit. 157 (2025) 110881.
[20] Y. Ma, et al., Class-imbalanced learning on graphs: a survey, ACM Comput. Surv. 57 (8) (2025) 1–16.
[21] J.C. Macuácua, J.A.S. Centeno, C. Amisse, Data mining approach for dry bean seeds classification, Smart Agric. Technol. 5 (2023) 100240.
[22] T. Feizi, M.H. Moattar, H. Tabatabaee, M2GDL: multi-manifold guided dictionary learning based oversampling and data validation for highly imbalanced classification problems, Inf. Sci. 682 (2024) 121280.
[23] L. Djafri, PRO-SMOTEBoost: an adaptive SMOTE boost probabilistic algorithm for rebalancing and improving imbalanced data classification, Inf. Sci. (2024) 121548.
[24] W.C. Sleeman IV, B. Krawczyk, Multi-class imbalanced big data classification on spark, Knowl. Based Syst. 212 (2021) 106598.
[25] M.H. Shirvan, M.H. Moattar, M. Hosseinzadeh, Deep generative approaches for oversampling in imbalanced data classification problems: A comprehensive review and comparative analysis, Appl. Soft Comput. 170 (2025) 112677.
[26] K. Borowska, J. Stepaniuk, A rough-granular approach to the imbalanced data classification problem, Appl. Soft Comput. 83 (2019) 105607.
[27] W.W. Ng, et al., Maximizing Minority Accuracy for Imbalanced Pattern Classification Problems Using Cost-sensitive Localized Generalization Error Model, 104, Applied Soft Computing, 2021 107178.
[28] M. Zheng, et al., An automatic sampling ratio detection method based on genetic algorithm for imbalanced data classification, Knowl. Based Syst. 216 (2021) 106800.
[29] Q.H. Doan, et al., A cluster-based data splitting method for small sample and class imbalance problems in impact damage classification, Appl. Soft Comput. 120 (2022) 108628.
[30] J. Ren, et al., Equalization ensemble for large scale highly imbalanced data classification, Knowl. Based Syst. 242 (2022) 108295.
[31] A. Dixit, A. Mani, Sampling technique for noisy and borderline examples problem in imbalanced classification, Appl. Soft Comput. 142 (2023) 110361.
[32] M. Pepsi, B. Binolin, N.S. Kumar, Hybrid firefly optimised ensemble classification for drifting data streams with imbalance, Knowl.BASED Syst. 288 (2024).
[33] S.A. Alex, J.J.V. Nayahi, S. Kaddoura, Deep convolutional neural networks with genetic algorithm-based synthetic minority over-sampling technique for improved imbalanced data classification, Appl. Soft Comput. 156 (2024) 111491.
[34] T. Liu, et al., Federated learning enabled hotel customer classification towards imbalanced data, Appl. Soft Comput. 166 (2024) 112028.
[35] S. Kamro, M. Rafiee, S. Mirjalili, Metaheuristic-driven space partitioning and ensemble learning for imbalanced classification, Appl. Soft Comput. 167 (2024) 112278.
[36] D. Singh, B. Singh, Investigating the impact of data normalization on classification performance, Appl. Soft Comput. 97 (2020) 105524.
[37] T. Le, et al., A hybrid approach using oversampling technique and cost-sensitive learning for bankruptcy prediction, Complexity 2019 (1) (2019) 8460934.
[38] G. Ke, et al., Lightgbm: a highly efficient gradient boosting decision tree, Adv. Neural Inf. Process. Syst. 30 (2017).
[39] I. Roman, et al., Bayesian optimization for parameter tuning in evolutionary algorithms. 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2016.
[40] Mukherjee, A., et al., Block-based Local Binary Patterns for Distant Iris Recognition Using Various Distance Metrics.
[41] J. Coronel-Reyes, et al., Multiclass classification of dry bean grains using machine learning techniques. International Conference on Technologies and Innovation, Springer, 2024.
[42] V.H.S. Lopes, et al., Performance evaluation of data analysis techniques in dry bean seed classification using kNN and MLP. Ibero-American Conference on Artificial Intelligence, Springer, 2024.
[43] P.V. de Campos Souza, E. Lughofer, An interpretable uni-nullneuron-based evolving neuro-fuzzy network acting to identify Dry Beans. 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2022.
[44] M. Koeshardianto, et al., Classification of dry-beans using synthetic minority over-sampling technique and stochastic gradient boosting machines. AIP Conference Proceedings, AIP Publishing, 2024.
[45] J. Nayak, P.B. Dash, B. Naik, An advance boosting approach for multiclass dry bean classification, J. Eng. Sci. Technol. Rev. 16 (2) (2023).
[46] S. Krishnan, et al., Identification of dry bean varieties based on multiple attributes using catboost machine learning algorithm, Sci. Program. 2023 (1) (2023) 2556066.
[47] B. Dejene, G. Setegn, S. Belay, Explainable and interpretable dry beans classification using soft voting classifier. Proceedings of the Data Science Agriculture Africa, DAAfrica, Algeria, 2025, p. 24.
[48] Vaidya, H. and K. Prasad, Multiclass Classification of Dry Beans using Artificial Neural Network.
[49] C.-Y. Lee, W. Wang, J.-Q. Huang, Clustering and Classification for Dry Bean Feature Imbalanced Data, Sic. Rep. 14 (1) (2024).